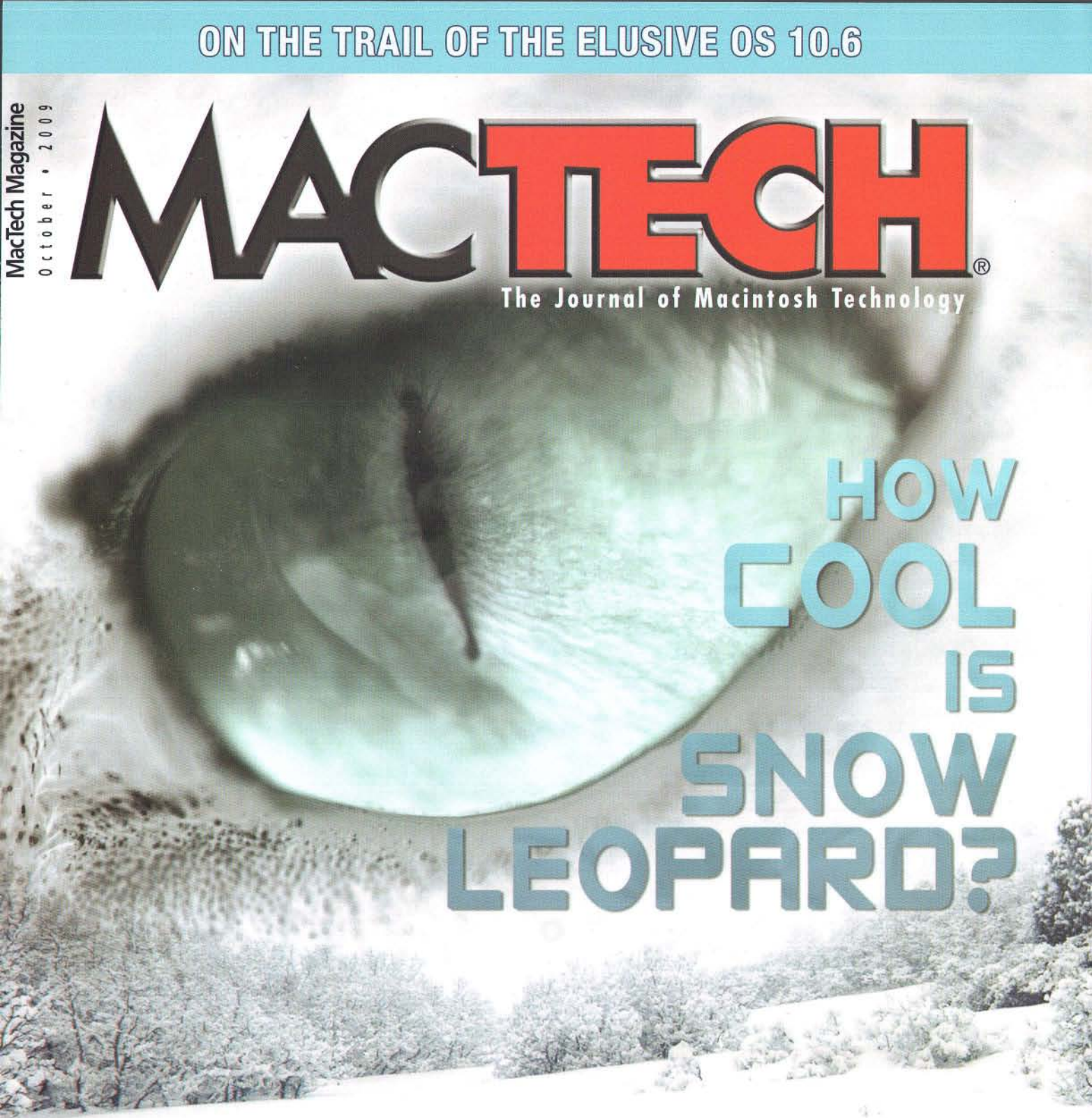
ON THE TRAIL OF THE ELUSIVE OS 10.6

MacTech Magazine
October • 2009

MACTECH®

The Journal of Macintosh Technology



HOW COOL IS SNOW LEOPARD?

MACTECH.COM

$8.95 US, $12.95 Canada

10

0 32128 74887 8

ISSN 1067-8360 Printed in U.S.A.

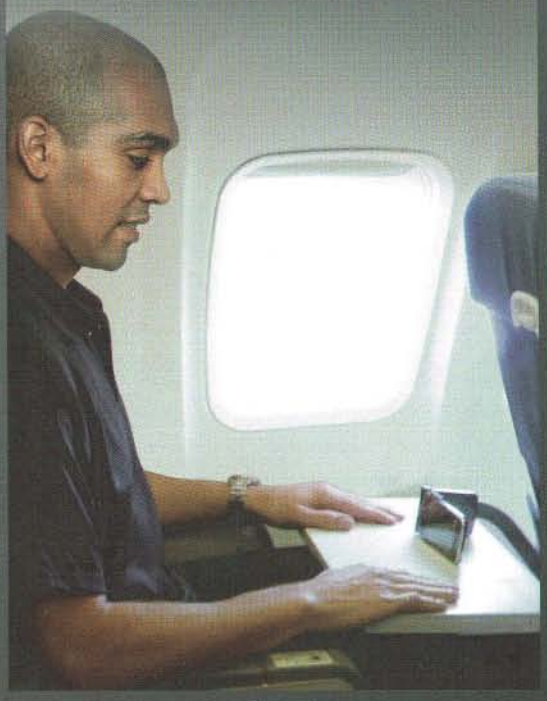
dexim™ *Put your dreams in life.*

www.dexim.net

P-Flip™
Foldable Power Dock

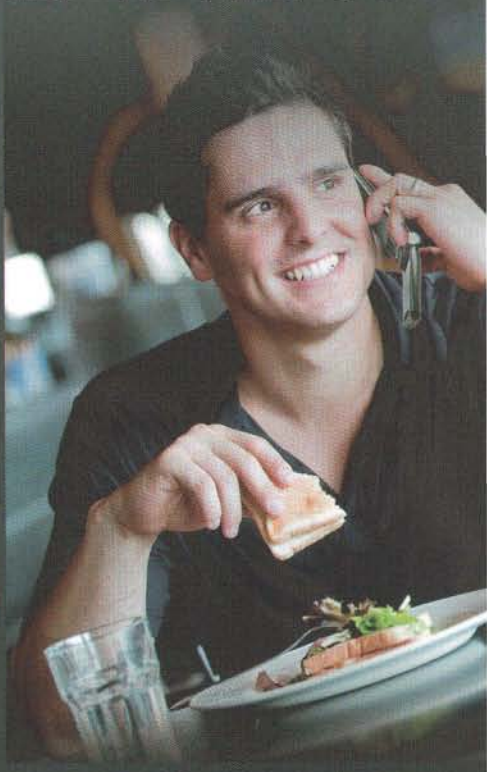
The Perfect pocket solution for your iPhone 3GS/3G/iPod touch

▾ UP TO **15HRS** GAMES/
60HRS MUSIC



▾ UP TO **15HRS** VIDEO

Lithium Polymer
BATTERY



▾ UP TO **8HRS** TALK TIME

(2.6"× 3.05"×0.66")

PORTABLE, FOLDABLE
RECHARGEABLE

3G S | 2000 mAh

Charge & Sync for
iPhone 3GS/3G/iPod touch 3G/2G/1G

M B S G P

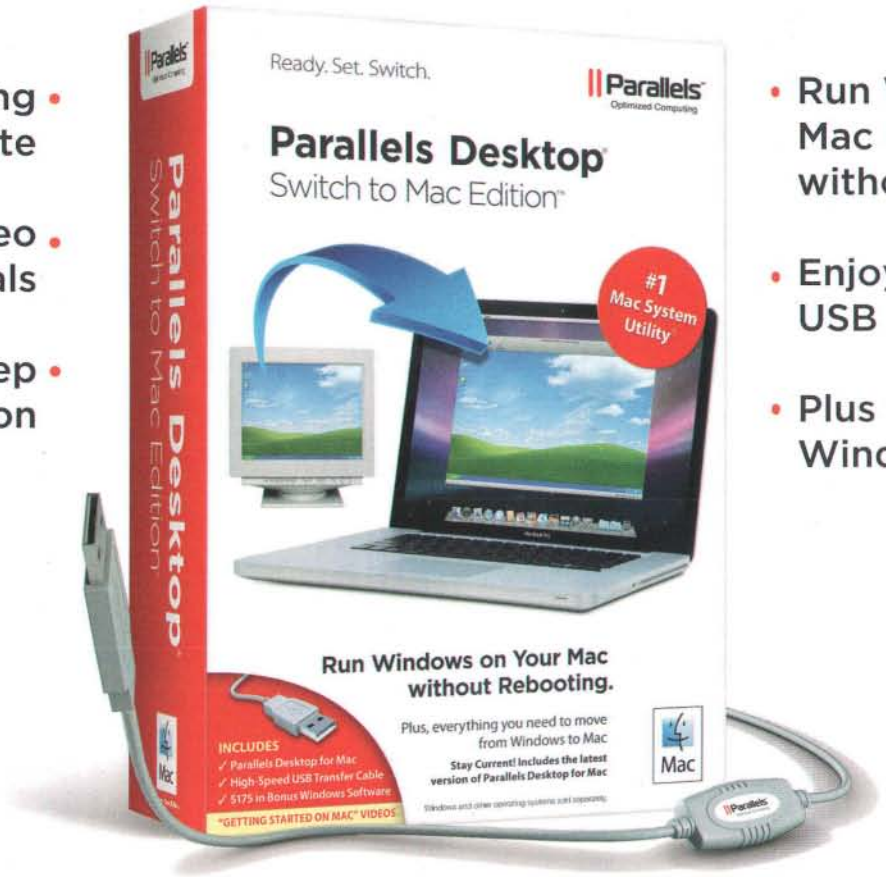
NEW

Switch to Mac
has never been easier.

Parallels Desktop® Switch to Mac Edition

Ready. Set. Switch.

- Complete Moving Suite
- Interactive Video Tutorials
- Step-by-Step Easy Migration



- Run Windows and Mac Side-by-Side without Rebooting
- Enjoy Your Favorite USB Devices
- Plus $175 Bonus Windows Software

Parallels Desktop Switch to Mac Edition lets you move programs, documents, media and more right from your PC to your new Mac. Then enjoy the best of both worlds and run Windows and Mac OS X side by side.

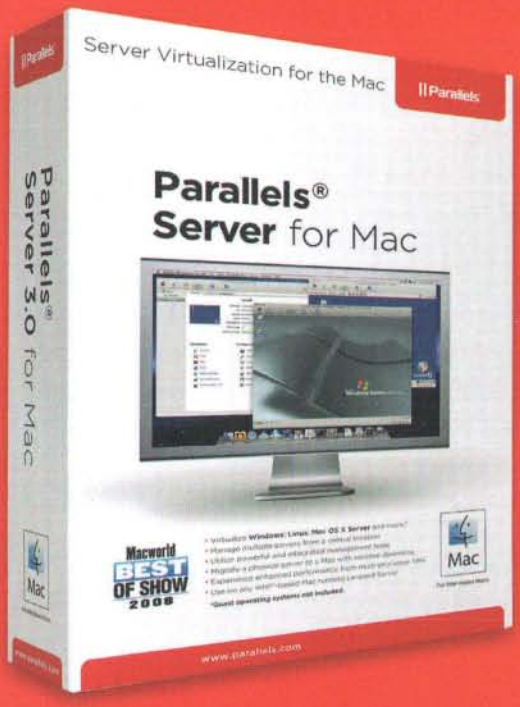
Learn more at
www.parallels.com/products/desktop/stm

Parallels
Optimized Computing

Phone +1 (425) 282-6448

Parallels™

Parallels® Server for Mac



Make your Mac Server go farther.

Run any OS you choose.

Run any application on the Apple Xserve today.

The world's first server virtualization solution for the Mac platform.

Reduce Server Count by 84%

Save 60% on IT Budget

Oregon City School District

Performance

Full scale hypervisor solution with bare metal architecture.

Scalability

Virtualize Mac OS X Server, Windows, Linux and more with support for 32- and 64-bit platform and guest OS support.

Flexibility

Hardware-ready for seamless integration into existing IT infrastructures with built-in VM management and maintenance tools.

Buy now at www.parallels.com or email offer@parallels.com

Business is about Accomplishment. ConceptDraw Office is all about Business.

Strategic Accomplishment

To accomplish something, is to bring about a desired result. The individual components of accomplishment can be broken down as either tactical or strategic.

Tactical is often viewed as small-scale actions that contribute to a larger purpose, while strategic accomplishment is more focused on the integrated planned effect of a company. Strategy comes in many shapes; likewise there can be varying levels of impact. The one given is that attaining a successful strategy is critical to any business.

A Strategic Accomplishment Map can be used for keeping your strategy visible at all times. This example **http://conceptdraw.com/accomplish** may be used as a template. A company's strategic plan must be broken down to departmental and individual levels. If this is not done, the company's strategy may never be effective, as participants will not understand their roles.

"*If the building blocks are clearly defined, the essence of even the most complex strategy can be expressed on one page.*" -An excerpt from, Execution. The Discipline of Getting Things Done.

At CS Odessa our version of this quote is: "*If the building blocks are clearly defined, the essence of even the most complex strategy can be expressed on ONE MIND MAP.* "

Mapping Strategy Implementation

Many times strategy, after the initial planning stage, is elusive in its identification and implementation. Mind mapping is a great starting point to sort through this, as many of our customers use ConceptDraw Office to assist them in attaining their goals. ConceptDraw Office is a business solution that combines business productivity methods and data visualization with project management technologies. There is no other product on the market equipped enough to enable one to observe and analyze the accomplishment processes inside and out. ConceptDraw Office makes it a snap to plan, develop, manage, and document quality processes.

Seamless Software Solution

ConceptDraw Office is comprised of ConceptDraw MINDMAP, ConceptDraw PROJECT, and ConceptDraw PRO and is packaged as one seamless software tool kit.

ConceptDraw MINDMAP

ConceptDraw MINDMAP is an excellent tool to record observations and development of goals. Moving from observation stage to early process development is as easy as dragging-and-dropping items or objects that can be used to structure a process and then immediately assign resources.

ConceptDraw PROJECT

With a single push of a button, all observations and task information are automatically moved over from ConceptDraw MINDMAP to ConceptDraw PROJECT. From a project vantage point it is then easy to manage all the resources available to you and ensure proper management of all the processes put into place.

ConceptDraw PRO

ConceptDraw PRO is a professional business graphics and drawing tool that is an integral part of ConceptDraw Office. ConceptDraw PRO is well known for its ability to produce eye popping business graphics and flow charts in professional formats, and to build score card documents that report on project status.

Plan, Do, Communicate. Accomplish.

ConceptDraw Office is a proven and powerful software solution that can assist you in the common challenges you are faced with when working towards accomplishing your company's defined goals.

Works Cited

Bossidy, Larry, Ram Charan, and Charles Burck. Execution The Discipline of Getting Things Done. New York: Crown Business, 2002.

Hunger, J. David, and Tom Wheelen. Essentials of Strategic Management (4th Edition). Upper Saddle River: Prentice Hall, 2006.

A fully functional 30-day trial copy of ConceptDraw Office may be downloaded for free by visiting: http://www.conceptdraw.com

CSO Corp.
Odessa, Ukraine
Sale/Support: +38 048 760-1963

CS Odessa LLC
Americas
Sales/Support +1 877 441-1150/ +1 408 441-1150 ex 4

TABLE OF CONTENTS

Mac in the Shell

Snow Leopard in the Shell

Changes to the shell environment in OS 10.6

by Edward Marczak 8

Humane Project Planning

All we know about starting an Apple business, from the idea, to product launch, and beyond.

by Michael Göbel and Oliver Pospisil, Inspired By Life 14

Unlocking Podcast Producer 2

by Michele (Mike) Hjörleifsson 30

MacEnterprise

Snow Leopard, Launchd, and Lunch

More launchd recipes, and a look at changes in Snow Leopard

by Greg Neagle 38

Apple Script in Snow Leopard

by Ben Waldie 50

The Road to Code

Look What the Cat Dragged in

What's new in Snow Leopard

by Dave Dribin 60

MacEnterprise

Packaging for Enterprise Deployment

Building Installer packages for mass unattended distribution

by Greg Neagle 70

Kool Tools

BookEndz

by Dennis Sellers 74

Kool Tools

Catching Up with Screen Capturing

Camtasia for Mac, Voila, ScreenFlow, and SnapzPro

by Dennis Sellers 76

The MacTech Spotlight

Jesse Grosjean

Hog Bay Software 80

From the Editor

Welcome to MacTech Magazine's Snow Leopard issue. We're really excited about Snow Leopard and this issue pulls together great content showing the diverse new features of Apple's new release. While Apple is reporting large sales of the new OS, we don't know that many people who have actually committed to it as a production OS. But we're sure of two things: Apple will ship new hardware that only runs 10.6. Additionally, for those with an existing machine, or machines, there will be some compelling reason to upgrade at some point.

Developers: look to Dave Dribin's Road to Code column to begin to understand the changes that Apple offers the programmer. Developers will certainly be driving people to upgrade, as some of the new 10.6-only features are too compelling to ignore.

Mike Hjörleifsson brings us part 1 of a look at the new Podcast Producer version 2. Frankly, even I hadn't had the chance to dig into the newest version shipping with Snow Leopard. Mike's intro made me want to get a closer look, and if you need to process any media, you'll be excited by the possibilities, too. Let Mike get you started with Podcast Producer in Snow Leopard.

MacEnterprise follows up on last month's launchd recipes with updates to launchd in 10.6. This is a must read for Sys Admins and everyone that uses launchd (so, this means *you*). We're also running a MacEnterprise article that deals with best practices in creating packages for distribution – this touches on Sys Admin and developer topics, so, this is another must read.

This month's Mac in the Shell gets back to its roots: what changes to the shell environment has Snow Leopard brought? Well, it's more than first meets the eye. "Snow Leopard in the Shell" has more.

Ben Waldie returns to illustrate the many changes Apple has brought to AppleScript in 10.6. It was a bit surprising to learn of the new inclusions to the language *and* what is going away. Ben's deep AppleScript experience informs this article and helps you to not get caught unaware.

There's also more our popular columns that aren't 10.6-specific, like Kool Tools, Michael Göbel and Oliver Pospisil's Inspired by Life series and more.

This month's MacTech Spotlight features Jesse Grosjean of Hog Bay Software. You'll find wonderfully simple, clearly thought out software at Hog Bay. Get a glimpse of the person behind the product.

Our Snow Leopard issue has been really enjoyable to produce—we hope you enjoy it too and that it clears the path for you. If you have topics that need clearing up, let us know: letters@mactech.com. Until next month, enjoy!

Edward Marczak,
Executive Editor

Communicate With Us

Department E-Mails

Orders, Circulation, & Customer Service
cust_service@mactech.com

Press Releases
press_releases@mactech.com

Ad Sales
adsales@mactech.com

Editorial
editorial@mactech.com
(Authors only, no pr)

Accounting
accounting@mactech.com

Marketing
marketing@mactech.com

General
info@mactech.com

Web Site
http://www.mactech.com

In this electronic age, the art of communication has become both easier and more complicated. Is it any surprise that we prefer **e-mail?**

If you have any questions, feel free to call us at 805/494-9797 or fax us at 805/494-9798.

If you would like a subscription or need customer service, feel free to contact MacTech Magazine Customer Service at 877-MACTECH

We love to hear from you! Please feel free to contact us with any suggestions or questions at any time.

Write to letters@mactech.com or editorial@mactech.com as appropriate.

MACTECH®

The Journal of Macintosh Technology

A publication of XPLAIN CORPORATION

Canada Post: Publications Mail Agreement #41513541
Canada Returns to be sent to: Bleuchip International, P.O. Box 25542, London, ON N6C 6B2

MacTech Magazine (ISSN: 1067-8360 / USPS: 010-227) is published monthly by Xplain Corporation, 5776-D Lindero Canyon #189, Westlake Village, CA 91362. Voice: 805/494-9797, FAX: 805/494-9798. Domestic subscription rates are $47.00 per year. Canadian subscriptions are $59.00 per year. All other international subscriptions are $97.00 per year. Please remit in U.S. funds only. Periodical postage is paid at Thousand Oaks, CA and at additional mailing office.

POSTMASTER: Send address changes to **MacTech Magazine**, P.O. Box 5200, Westlake Village, CA 91359-5200.

by Edward Marczak

Snow Leopard in the Shell

Changes to the shell environment in OS 10.6

Welcome

Snow Leopard is out and, like many, you've either upgraded, will upgrade or know someone who has. Billed as a release with "no new features," we all realize that's not quite the case. Most changes, rather than visual are 'under the hood.' And isn't that what readers of this column are interested in?!? This month, we'll take a look at changes to, and utilities for, the shell environment in 10.6.

Terminal.app

The logical place to start is with Terminal.app, Apple's interface to a command shell. Still located in /Applications/Utilities, Terminal.app is listed as version 2.1. Version 2.0 which arrived with Leopard was a very nice step up from earlier versions. Version 2.0 brought us a tabbed interface that allowed multiple shells in one tabbed window. The Snow Leopard version is, ummmmm, boring—in a good way! This isn't a tool that we want dramatic changes on with each release of the OS. We do officially get split pane terminals in version 2.1.

What's a split-pane terminal? Just as Leopard gave us multiple tabs in a single window, Terminal.app v2.1 in Snow Leopard gives us multiple *panes* in a single tab. Each pane can be scrolled to a specific part of that window's session. Figure 1 illustrates the effect with two panes.

Apple's Snow Leopard enhancements page touts, "Command-line users can now split their window into different terminal sessions." It also says that, "All features of Snow Leopard are subject to change," and I think that's the category this falls under. While a split-pane terminal is nice, there doesn't actually seem to be a way to place a different *session* into each pane. I suspect I simply have a different definition of "session" than Apple's marketing department. While that may be true, what I do have is GNU screen, which *does* allow splitting a single window into multiple separate sessions. There's a MacTech article on screen on-line at http://www.mactech.com/articles/mactech/Vol.21/21.09/Screen/index.html.

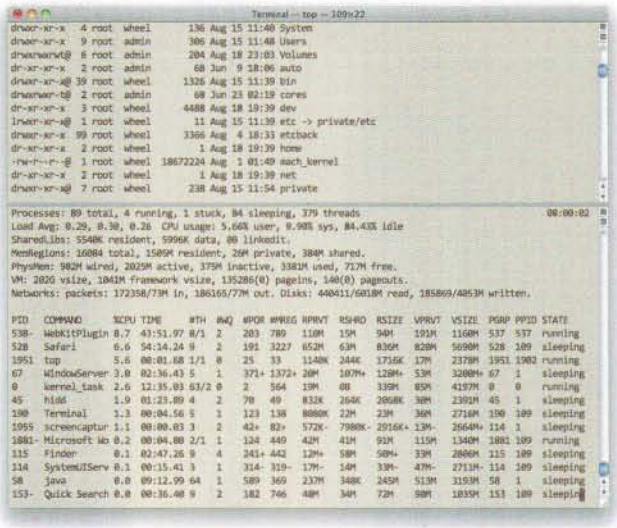


Figure 1: Terminal.app with split panes

Another nicety is the appearance of a new font: Menlo. Menlo is a fixed-width programmer's font. I found myself happily using it immediately. While one's favorite font is a bit subjective, I think you'll have Menlo in your top 3 or so programmer fonts pretty quickly. Figure 2 shows a sample of this new font.

```
Last login: Wed Aug 19 07:57:43 on ttys010
```

Figure 2-Menlo, the new default font in Terminal.app.

Menlo is a little lighter and tighter than Monaco or Andale Mono, allowing for easier viewing at smaller sizes. This is particularly noticeable on 13" screens such as the MacBook or MacBook Air. Menlo uses DejaVu Sans Mono as its base, with Apple tweaks from there. The most loved (or hated) tweak may be a dotted-zero. Most people I know want a slashed or dotted zero. Menlo gives you the dotted variety, as seen in Figure 2.

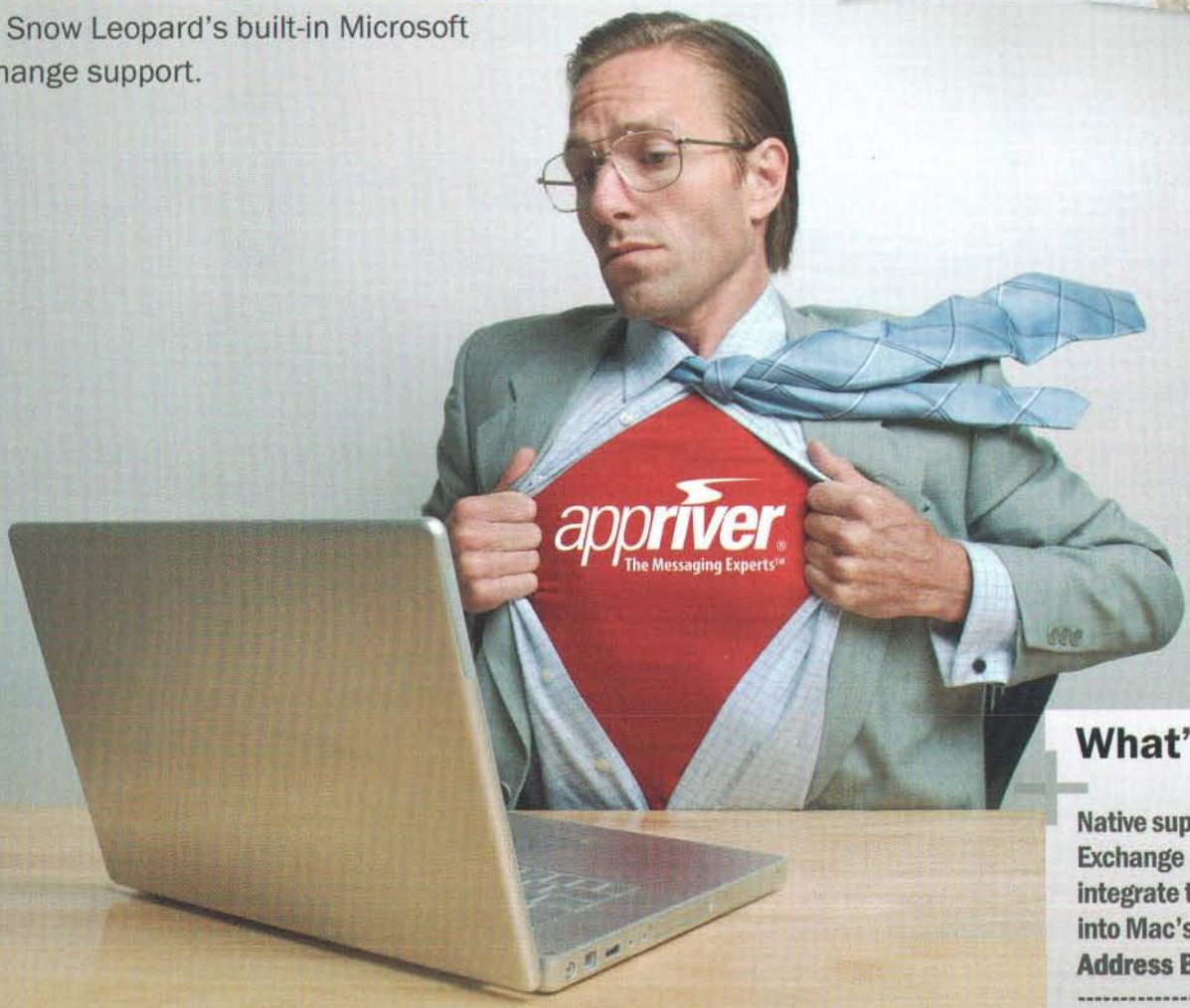
If you're in love with Monaco, it's still shipped with the system, never fear. However, Menlo now becomes the default for the "Basic" setting in Terminal.app and is also the default in Xcode.

The Shells

Of course, a terminal application doesn't do us much good if it doesn't have anything to control. The default shell, bash,

The Best of Both Worlds

Unleash the power of your Mac, with **AppRiver's Secure Hosted Exchange** and Snow Leopard's built-in Microsoft Exchange support.



What's New:

Native support for Microsoft Exchange allows a user to integrate their Exchange data into Mac's Mail, iCal and Address Book applications.

Unlimited Email Storage
No need to manage your mailbox storage quotas.

The Only Optimized Hosted Exchange Service
Akamai-enabled, providing consistent, reliable access to all of your mission-critical communication applications.

Better Mobility
Synchronize your iPhone with our FREE ActiveSync service.

www.appriver.com/mac
sales@appriver.com
(866) 223-4645

Snow Leopard is a registered trademark of Apple Inc.
Microsoft Exchange is a registered trademark of Microsoft Corporation.

doesn't see too much of a change. It's now at version 3.2.48(1)-release (up from 3.2.17(1)-release):

```
$ bash -version
GNU bash, version 3.2.48(1)-release (x86_64-apple-
darwin10.0)
```

Apple, following their own advice, have bash compiled as a 64-bit binary, as are all shell binaries I've found. The above output will only show if you're booted into a 64-bit kernel.

If you're a zsh or tcsh user, note that these shells have moved into /bin with bash, csh, ksh and sh. That's a notable change that may require you to update a script's shebang line.

For more detail on the changes in a version of bash, see the changes list at http://tiswww.case.edu/php/chet/bash/CHANGES (and yes, you'll notice that bash is currently at version 4.x).

Scripting Languages

Since this column has been focusing on Python for the last few months, it's worthwhile to note that most version numbers for scripting languages have been upgraded, too.

Python now uses version 2.6.1 as its default. Version 2.5 is also loaded on the system and ready. If you heard, at one point, that version 3.0 would ship with OS X 10.6, you were relying on pre-release information. Python 3.0 is not present.

It's worthwhile verifying that any Python scripts you rely on run properly in the new environment. This holds especially true in situations where you may use an ambiguous she-bang line (not specifying a particular version of Python). Like other binaries on the system, Python 2.6 receives the 64-bit treatment.

The python man page has some really good information (just ignore any references to python v3.0). First, you can easily choose which version of python you want to be the default. Writing a value to com.apple.versioner.python allows this:

```
defaults write com.apple.versioner.python Version 2.5
```

...will make version 2.5 the user default when running the both the python and pythonw commands. The environment variable VERSIONER_PYTHON_VERSION can also be used to set the python and pythonw version:

```
export VERSIONER_PYTHON_VERSION=2.5
```

You may also want to force python 2.6 to run in 32-bit mode, which is also easily achievable:

```
defaults write com.apple.versioner.python Prefer-32-Bit -
bool yes
```

...or, the the appropriate environment variable:

```
export VERSIONER_PYTHON_PREFER_32_BIT=yes
```

Of course, you may want to set the system-wide default, in which case, you should write into /Library/Preferences/com.apple.versioner.python, and/or set the environment variable in some global startup file.

For the record, Ruby sees a bump to v1.8.7 and perl to v5.10.0 in OS X v10.6.0.

Macports

MacTech has covered Macports and Fink in the past as ideal ways to expand your toolbox of applications and command-line utilities. I haven't tested Fink, but Macports released a new installer, version 1.8 along with the Snow Leopard rollout. If you're someone who builds from source, you may have not noticed anyway—installing from source worked cleanly though the beta seeds of 10.6, but the installer failed. Personally, I like the build-from-source method as I can customize the install.

Installing Macports from source is easy. Ensure you have the Developer tools loaded on your machine. Be a little more cutting edge by checking out the latest source from subversion, and install:

```
mkdir -p ~/src/macports
cd ~/src/macports
svn checkout
http://svn.macports.org/repository/macports/trunk/base
cd base
./configure -prefix=/path/to/install/opt/local/
make
sudo make install"
```

This can even be automated to a certain degree, which you may want to do if you need to install Macports in an adhoc manner. Otherwise, the Macports bin directory can be built on a master machine and then synced to clients that require the binaries.

Updates to Command-Line Apps

Several command line applications have been updated to provide better or more complete output. One that we've talked about in this column is the `top` utility.

```
Processes:  112 total, 2 running, 3 stuck, 107 sleeping... 438 threads
Load Avg:  0.29,  0.20,  0.21    CPU usage:  9.84% user,  9.43% sys, 80.74% idle
SharedLibs: num =   16, resident =   52M code, 1008K data, 3844K linkedit.
MemRegions: num = 17005, resident =  627M +   30M private,  193M shared.
PhysMem:  355M wired,  952M active,  450M inactive, 1757M used, 1315M free.
VM: 12G + 376M   374898(0) pageins, 39(0) pageouts
```

Figure 3 – default top output in 10.5

The banner section in top has a few tweaks that improve it for readability—no more equal signs and other parts labeled better. Additionally, the 10.6 version gives the additional information you'd normally go to netstat and iostat for: network and disk traffic. Of course, netstat and iostat give more detailed information when needed.

```
Processes: 94 total, 2 running, 92 sleeping, 343 threads
Load Avg: 0.52, 0.41, 0.35  CPU usage: 18.86% user, 6.60% sys, 74.52% idle
SharedLibs: 5124K resident, 6688K data, 0B linkedit.
MemRegions: 17759 total, 1579M resident, 32M private, 516M shared.
PhysMem: 858M wired, 2089M active, 539M inactive, 3486M used, 610M free.
VM: 201G vsize, 1037M framework vsize, 198083(0) pageins, 0(0) pageouts.
Networks: packets: 276078/171M in, 153392/15M out. Disks: 265840/4237M read, 9644
```

Figure 4 – default top output in 10.6

launchd sees many improvements in its 10.6 incarnation. Greg Neagle tackles the changes to launchd in this month's MacEnterprise column, so be sure to continue your reading there (if you haven't already).

The End?

Snow Leopard, OS X v10.6, brings much finer changes than any version of OS X before it. We'll likely see more refinements (and changes! So always be on the alert) as point releases are shipped.

Media of the month: For those of you that like a long, sweeping book, "Underworld" by Don DeLillo is a bit of a modern classic. It's well worth the investment in time.

Until next month, enjoy poking around Snow Leopard.

About The Author

Ed Marczak is the Executive Editor for MacTech Magazine, and has written the Mac in the Shell column since 2004.

Offered by Intego and these best-of-class security partners:
Micromat, Absolute Software, Macware, JoeSoft, Smith Micro, MOApp and GagaFactory

SPECIAL OFFER - ONE TIME ONLY

Protect your Mac with the ultimate security bundle at an AMAZING price!

Act now to take advantage of this one-time offer. This bundle is available until October 31, 2009.

Intego
VirusBarrier X5
The acclaimed antivirus program for Mac.

Micromat
TechTool Pro 5
Hard drive and system repair and maintenance tool.

Absolute Software
Computrace LoJack
Tracks, locates, and recovers stolen computers.

Intego
ContentBarrier X5
Keeps kids safe from the dangers of the Internet.

JoeSoft
Klix
Finds and recovers corrupted pictures on your camera or media card.

Intego
Personal Backup X5
Backs up files, synchronizes folders, and clones startup volumes.

Macware
WebGhost
Hides your IP address so you can surf anonymously.

Intego
NetBarrier X5
Protects your Mac from Internet attacks.

Intego
Personal Antispam X5
Keeps your inbox spam-free, and filters phishing e-mails.

MOApp
MyWallet
Secures passwords and other important information.

Smith Micro
Spring Cleaning
Organizes your hard drive and eliminates duplicate files.

Intego
FileGuard X5
Safeguards your confidential files in virtual safes.

Free bonus for the first 10,000 purchasers

GagaFactory
Radio Gaga
Listen to and record 10,000+ Internet radio stations on your Mac.

90% OFF
12 APPS FOR ONLY $49.99
$~~690~~
Mac OS X 10.6 Snow Leopard compatible

MACTECH "Mac users typically take security for granted. They shouldn't. This bundle gives users of all levels peace of mind and greater protection. And at an amazing price."
Neil Ticktin, Editor-in-Chief/Publisher, MacTech Magazine

ORDER NOW Not available in stores!

web www.macpromo.com
e-mail bundle@macpromo.com
phone (512) 637-0700

Offered by Intego and these best-of-class security partners:

Mac Universal

www.macpromo.com

Humane Project Planning

All we know about starting an Apple business, from the idea, to product launch and beyond.

by Michael Göbel and Oliver Pospisil, Inspired By Life

Inside Inspired By Life

Michael: "Oliver, I'm not satisfied with the outline view."

Oliver: "What bothers you about it?"

Michael: "I don't think so many users really need it."

Oliver: "What feature could be more important?"

Michael: "That's exactly the problem. Right now, it's just more of a gut feeling."

Oliver: "Ok. Let's tackle the problem from another side. We need an outline view to enable the user to put the elements into hierarchical order. Which feature would add more value for the user?"

Michael: "Initially, we had planned on offering attachments in version 1.x. So shouldn't we still do it? When I checked our decision log out again, it sure looks like both of us had very strong, positive arguments for the outline view and for the attachments, too."

Oliver: "Yes, we had opted for the outline view for one of our potential customers. Since we've now adjusted our overall strategy, let's think of a way to implement both."

Michael: "Ok, since another developer has teamed up with us, that should be doable. I'll get back to you on this as soon as possible."

Introduction

We will now be your guide during the following planning session. At the end of this article, you'll know what it takes to come up with a plan that's worth the time invested and that provides the right trigger for non-coding activities like a press release.

The Plan, reframed

"[Planning] is the last refuge of those who cannot dream," Oscar Wilde.

Most people perceive a plan to be something that makes them feel kind of guilty since plans generally do not end up being error-free. Others see a plan as a broken promise and nobody likes it when promises are not kept. Therefore: No planning, no broken promises.

This mental model needs to be reframed:

Imagine that you're climbing up a mountain. The path ahead of you is blocked and you need an alternative path to reach the top. What do you do? You pull out your map, look for alternative paths and decide on the best one to take.

The plan, while developing a software program, is your map. Not only is the plan a way to think through the details of your software in depth. It is also becomes the most valuable tool the minute you pinpoint a gap between your plan and actual reality: You pull your plan out to decide what the best course of action is to get back on track.

Triggering is the second reason for planning. The release of a software program entails far more than just pure coding. It involves a beta program, marketing and sales, i.e. non-coding activities. All activities must be in sync to reach the release date as soon as possible.

As a manager, for example, I need to know

When I have to have a private beta testing team set up,

When the content for the website needs to ready,

When the payment system needs to be in place,

When the help and support documentation needs to be ready and

When the press release must be published.

In software development, the coder sets the pace. The plan provides all non-coders with the right trigger to get their job done.

The Planning Session

Now we'll get the right tools in place and guide you through all of the planning steps.

A Minimalistic Toolbox: Numbers and Pen & Paper

The toolkit should be as lightweight as possible: We recommend Numbers (or MS Excel) to list all of the software's features and to make estimates. All you need for GUI prototyping is a pen and a stack of paper.

CAMTASIA

[PRONOUNCED cam•TAY•zha]

INTRODUCING THE ALL-IN-ONE SCREEN RECORDER DESIGNED FOR MAC

Record screen activity
Edit your content
Add effects
Share your video

Download the free trial and check out our introductory pricing today at: **www.camtasiamac.com**

Camtasia®:mac

TechSmith

You might tell yourself "I'm not a painter." However, you don't have to be a painter to perform GUI paper prototyping. By putting GUI prototypes down on paper, you think through your application's features in depth and it will take you less time than it would with any other tool.

You might say "But with pen and paper, I'll have to start all over again when I need to modify something." Yes, and this is for a good reason, too. Keep all of the different versions of your notes and occasionally spread them out next to each other to check the evolution (or revolution). Make sure to put the date and time on each piece of paper.

While conducting GUI prototyping on paper you will find out that some forms stabilize or are predetermined (like the iPhone's display screen). Make a stencil out of it. As an example, a link to an iPhone stencil is included in the reference section of this article.

This is how Cultured Code did it:

Figure 1: iPhone stencil by Cultured Code

What if you're really different: You deliver the best results by implementing them right away. Well, if this is the case, take the liberty to do it your way.

An Enhanced Toolbox: Merlin and FogBugz

If your project is extremely complicated with highly complex dependencies use Merlin, which is the one and only project management tool that I would use in such a case. For all others (the majority) Numbers or Excel will do the trick.

There is one application specialist for software developers who work in remote areas: It's FogBugz. FogBugz is a bug tracker, project management tool and so much more; implemented by programmers for programmers – and not for managers. Since we now have a third team member on board (Michael discovered a great developer. Hi, Raphael!), we're now working in three different places and we set FogBugz up to update the plan, and it serves as a central hub for our seamless communication.

The Process: Initial setup and keeping it up-to-date

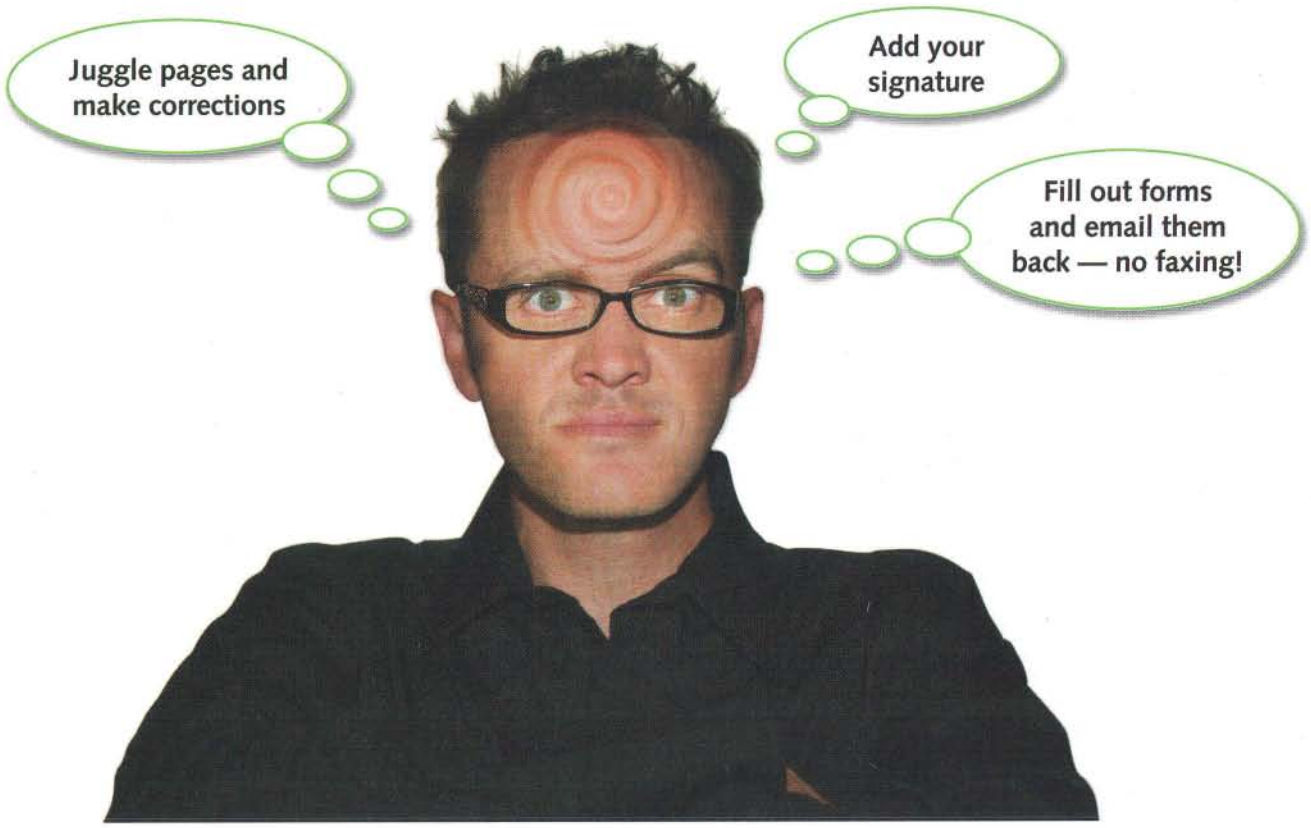
Now that our tools are in place, it's time for us to take action.

In essence, project planning is easy (often the tools or the method are what make it complicated):

First you need to brainstorm about all of your application's potential features, and then you need to list all of the features that must be in the first or next release and store all the rest for a future release. Finally, make sure to estimate the time it will take to implement each feature.

Secondly, update your plan. And that's all there is to it!

Achieve total domination over your PDFs



PDFpen™

The easy way to make PDFs bend to your will

Download a trial version of PDFpen today…
www.smileonmymac.com/mactech

Site licenses available starting at $999

Smile
on my mac

disclabel PDFpen pagesender textexpander

The First Meeting: Set the direction and get into the user's shoes

The first meeting for the planning session sets two main goals: The first is to get everyone involved to look ahead in the same direction and the second is to generate as many feature ideas as possible from as many viewpoints as possible: Go for quantity!

Get everyone who is involved in making sure the product is a success into the same room. It's the manager's job to get everyone to look in the same direction: Tell everyone a story dealing with the spirit of the soon-to-come software program. Then schedule a ten minute coffee break to get everyone to talk about it.

The second part will fill up the rest of the day: Let's get into the user's shoes!

The first thing you have to do is to come up with some scenarios on how the user will deploy the soon-to-come application based on the "One day in the user's life" stories. Ideally, you should draw small graphics like a comic strip. Come up with two to five scenarios.

Then find real world metaphors users would apply in the scenarios if they had to do it without a computer. Image life without Google maps: Take a real map and some pins to mark the direction from San Francisco to Houston. In addition, the user might have taken some notes on important points to remember.

Resist the temptation to talk about "What will it look like in the software." That's something that can wait until tomorrow.

Document everything on a flip chart, whiteboard, post-it notes or whatever suits your brainstorming session the best.

After the brainstorming is over, it's the manager's job to document and structure all the ideas.

The Second Meeting: Set the plan up

The next meeting on day number two will answer the question: "What will it look like in the software?" This is the right time to dig deeper and get specific. Don't be surprised if it takes you much longer than one day to finally answer this question.

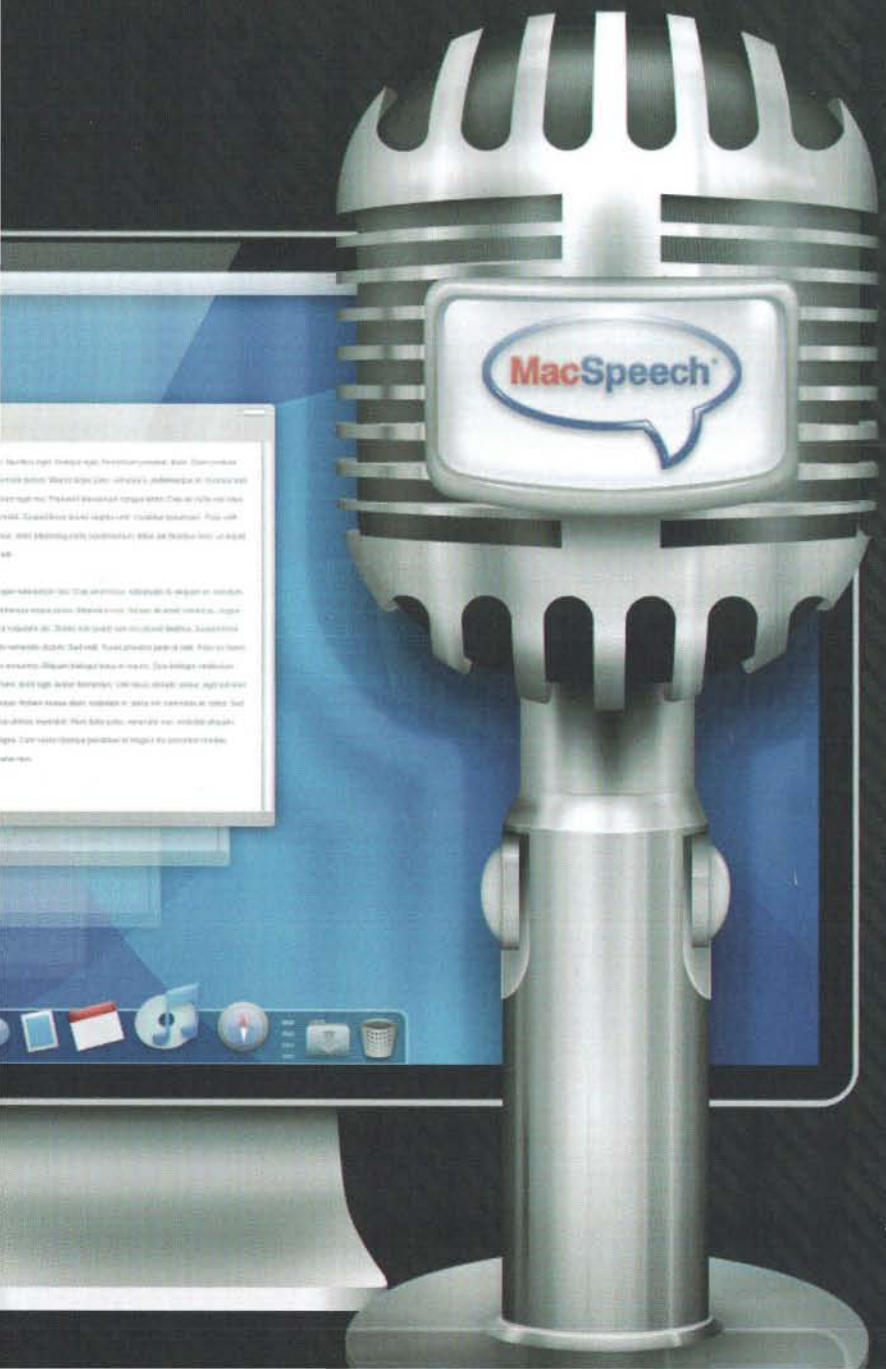
And time for action: Take the first scenario and the real-world metaphors and jot down GUI paper prototypes to transform it into software. Take notes to describe the non-visible functions. If possible, do this with different groups of people who are working on the same scenario.

Compare the ideas and interlink the best ones. Once you have a lot of GUI paper prototypes, a pattern of features will emerge. That's exactly what we want. It's the first indication that it is stabilizing.

To get an idea of what a GUI paper prototype looks like, here is the one for Things for iPhone by Cultured Code:

~~Type~~. Talk!

Premier speech recognition for Mac OS X.



MacSpeech®
Dictate

www.macspeech.com

Available from MacSpeech, Apple, and other fine Macintosh retailers.
Visit the MacSpeech website for a complete retailer listing.

Figure 2: GUI paper prototype, Things for iPhone by Cultured Code

Now do the same thing with all of the scenarios. Next, put the best solutions for each scenario next to each other for comparison. Again, check the pattern carefully and go for consistency: Buttons that trigger the same function must be in the same place and similar features should work in the same way.

One tip: Don't let a domain expert participate. Become a domain expert yourself and find your own solutions. For most non-software people it's almost impossible to imagine what things actually look like in software. Domain experts will come into play once you have a final-product-like-looking prototype.

Now it's time to decide which features to include in version 1.0. Put on the manager's hat and list all the must-have, super important features and set their priority to two. All of the nice-to-have features should be prioritized with a four or less. Et voilà: You have your feature specification.

You'll find some features in every good Mac software: Help, Icon, Website, Shop, Auto update, Serial number generator, Press release, ... put them on the feature list, too.

Now the ball goes back to the developer: Estimating how long it will take to implement each feature. To all the developers out there: Remember, you are doing this because the non-coders need to be triggered.

A personal note: Since you're not writing specifications for upper management, it's okay to have some fun. Name the user "Polly the potato." Write in a way so that your mom will understand the specifications, and not the compiler. It's typical to review and rewrite your specs several times.

Figure 3 (*below*) is part of the feature specs for our application:

How a good feature spec is structured

Here are some tips on how to write good feature specs:

- Write to get attention, and not to put people to sleep.
- Write to be understood.
- Write short active sentences not long or passive ones.
- Assign someone (and only one!) who is responsible for each spec.
- Describe only the invisible parts of the feature.
- Give every feature a title and a unique ID that never changes.
- If you use Pages or Word, make sure the whole document has a version number.

To find out more about how to write feature specifications, Joel Spolky's articles are an excellent source.

Regarding estimation

Rule of thumb: It takes three (!) times longer than you think.

This is why:

First, it takes a lot longer to implement a feature in a product because you need to get it ready for numerous exceptions to the rule. It's not like internal development where you can more or less coerce the user into using a feature in a certain way. Your customers will use it however they want and you must deal with their specific needs in a constructive way.

Second, getting the application released fuels your ideas further. Thus, your subconscious seduces you to underestimate the effort. That's okay it's human nature.

Be realistic and include the following in your estimation: you come down with the flu (five days each year), you take vacation (six weeks each year), time for debugging, and on the list goes.

Don't estimate features. They're just too complicated. Features consist of multiple functions. This is where the work for developers really begins: Thinking in great detail about which functions must be implemented to create a feature. In this way, developers think it over in great detail and they obtain an excellent baseline to estimate each function, before even starting to write one single line of code.

The scale estimation is in hours and not days or weeks. To all the managers out there: If you come across an estimation that is more than eight hours, ask the developer to rework it in sections consisting of less than eight hours. If an estimate is over eight hours, that definitely means the developer didn't think through it in enough detail.

To all managers: Whatever you think, the developer's estimation is authoritative. Period.

Inspired by Joel's famous Excel sheet for estimation, Figure 4 (*below*) is the one used for our application:

Feature	ID	Description
Outline View	17	No additions to the GUI paper prototype.
Attachments View	5	In addition to the GUI paper prototype: The attached files should not be stored directly in the application. Instead the files will be stored in a hidden folder on the HD and linked in the application.

Figure 3: Example for a feature specification

Done	ID	Feature	Function	Priority	Estimation	Elapsed time	Remaining	Trigger Management
☑	17	Outline View	implement indentation		4	7	-3	nothing to do
☐	17	Outline View	implement outdentation		4	2	2	nothing to do
☐	17	Outline View	Swap elements		6		6	Tell Oliver to schedule meeting with lead user
☐	17	Outline View	collapse and expand		8	0	8	nothing to do
☐	17	Outline View	Trigger save		1	1	0	nothing to do
☐	5	Attachments View	Attach droped file		4	0	4	nothing to do
☐	5	Attachments View	list attached file in attachment view		2	0	2	nothing to do
☐	5	Attachments View	create link to attached file		2	0	2	Tell Oliver to schedule meeting with lead user
☐	5	Attachments View	detaching file by drag & drop outside window		3		3	nothing to do
☐	5	Attachments View	Display file icon		1		1	nothing to do
☐	5	Attachments View	Trigger save		1		1	nothing to do

Figure 4: Example of feature and function planning

To find out more, read Joel Spolky's original and updated articles which zone in on estimations.

Setting priorities and making decisions

Quite likely, you've listed too many features for version 1.0. So now it's time to prioritize the features and decide which ones to incorporate in version 1.0.

The key rule: Features should only be ranked up to a maximum of priority #2. Priority number 1 is to be set for bugs and nothing else.

Take a close look at the scenarios and only rate those features with a priority #2 that are absolutely essential. All must-have features required by each application (incl. the user manual, support, updating service,) should be rated with a priority # 2 as well. Everything else should be prioritized between the range of 3 and 5.

Figure 5 (below) is a sample estimation sheet, including the prioritization.

Decision-making is almost always a hard thing to do. Even though the final outcome will simply be a Yes-or-No, you must nevertheless take your decision very carefully. Many useful approaches are available today on how to make the best possible decisions, however, you should only put one of them into actual practice. What works best for me is the approach crafted by Spencer – the "Yes-or-No Strategy." The following is a condensed version of the core components of his approach:

Step one: Avoid indecision and half-decisions based on half-truths.

Step two: After you thought deeply about your own rationale and have listened to opinions presented by others, you make a better decision and act on it immediately.

Step three: Is it necessary to decide and are you ready to do it?

Are you meeting a real need? - Is it a mere want or a real need?

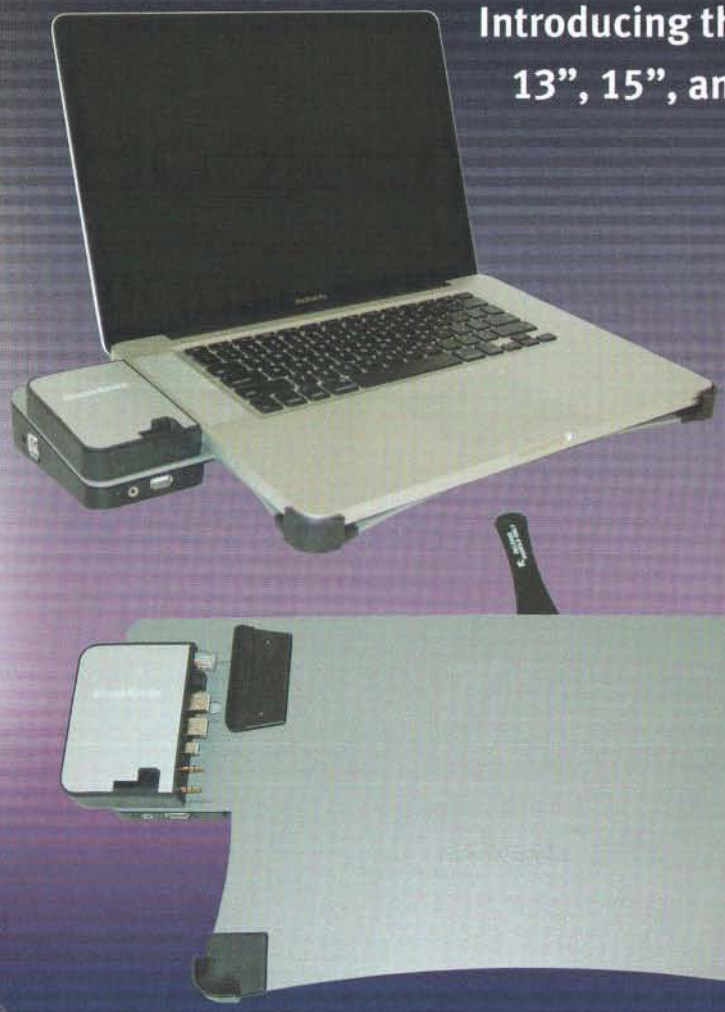
Done	ID	Feature	Function	Priority	Estimation	Elapsed time	Remaining	Trigger Management
☑	17	Outline View	implement indentation	2	4	7	-3	nothing to do
☐	17	Outline View	implement outdentation	2	4	2	2	nothing to do
☐	17	Outline View	Swap elements	2	6		6	Tell Oliver to schedule meeting with lead user
☐	17	Outline View	collapse and expand	4	8	0	8	nothing to do
☐	17	Outline View	Trigger save	2	1	1	0	nothing to do
☐	5	Attachments View	Attach droped file	2	4	0	4	nothing to do
☐	5	Attachments View	list attached file in attachment view	2	2	0	2	nothing to do
☐	5	Attachments View	create link to attached file	2	2	0	2	Tell Oliver to schedule meeting with lead user
☐	5	Attachments View	detaching file by drag & drop outside window	2	3		3	nothing to do
☐	5	Attachments View	Display file icon	3	1		1	nothing to do
☐	5	Attachments View	Trigger save	2	1		1	nothing to do

Figure 5: Example of estimation and prioritization

BookEndz® Docking Stations for Apple Computers

DOCKING STATIONS FOR APPLE COMPUTERS

**Convert your MacBook Pro® or MacBook®
into a desktop in seconds
without misplacing cables or damaging connectors.**



Introducing the New Docking Stations for the 13", 15", and 17" Unibody Apple Laptops

- Aluminum Plate helps in cooling of notebook
- Connectors are routed to back of Docking Station
- 5 USB 2.0 compliant ports
  (4 port powered or unpowered hub, AC/DC adapter included)
- Gigabit Ethernet RJ45
- Convenient front mounted Headphone and USB jack
- Security Lock Accessible

Visit our website for latest product announcement www.BookEndzdocks.com

BookEndz®
Manufactured by Olympic Controls
1250 Crispin Drive • Elgin, Illinois 60123 • USA
Phone: 847-742-3566 • Fax: 847-742-5686 • Toll Free: 888-622-1199
E-mail: Sales@BookEndzdocks.com

Are you informing yourself about the options? - What information do you need? Have you come up with feasible alternatives?

Are you thinking it through? - If you decided on "x", what would happen then? And then what?

Yes or No?

Step four: Do you remain true to yourself?

Are you being honest with yourself? - Are you telling yourself the truth?

Do you trust your gut intuition? - Does it feel right? You weren't afraid?

Do you deserve more? What would you do if I deserved better?

Yes or No?

Finally: If the answer is Yes, act on it. If the answer is No, work through it again.

Procrastination

"Action is the last refuge of those who cannot dream" by Oscar Wilde.

Do you feel a strong reluctance within yourself to list the features and estimations or to take the necessary decisions? If so, congratulations, this just goes to show you're in excellent mental shape. Procrastination is your inner shield that is there to help you sidestep failure. Nobody, of course, wants to see their all-out endeavors backfire and then ultimately fail.

What you do know is that to make sure your application is released, you cannot stop now. You must make feasible estimates, list the core features and take decisions, even when you run a risk of failing. The following are some ways that will help you succeed rather than fail:

1. Fully understand that something deep within you just wants to keep you out of harm's way.

2. Everything starts by taking the first step. So just do it! What you do doesn't have to be perfect. (I had to rewrite this article at least five times and in certain areas, revisions were made at least ten times. But that's how to do it and it does get done! Michael, for example, uses innumerable GUI prototypes to come up with the one that works best.)

3. Talk about it with others and ask for their help.

It's important to move forward everyday, at least a little bit. Taking a break intentionally is another way to move forward because you are tanking up on energy, reenergizing your batteries. However, if your break turns into a whole week instead of just a few hours, that could be a sign of procrastination.

Step two: Keep your plan up-to-date

An outdated plan is as useless as an old map. Keeping your plan up-to-date is a routine that you should turn into a habit at least once a week – and preferably once each day.

In this way, you will know when the triggers for non-coding activities are released and you will learn lessons crucial for the future.

Before unveiling version 1.0 on the market, it will not be a critically risky move to announce the release date later than

• FREE Sample Packs • FREE Business Cards • FREE Mail Lists •

The Nation's #1 Source for Full Color Printing & Direct Mail Marketing!

PrintDirectforLe$$.com

If interested in becoming a RESELLER/BROKER please contact us at sales@PrintDirectforLess.com

Up to 35% OFF list pricing

24 hour turnarounds available on most products

EARN A GREAT 2nd INCOME!

Full Color Printing

Direct Mail Marketing & Mail Lists

800.301.8870

PRICING • Full Color Printing

(4/4) 4" x 6" POSTCARDS	
500	$ 69
1,000	$ 80
2,500	$119
5,000	$182
10,000	$307

(4/4) 8.5" x 5.5" POSTCARDS	
500	$129
1,000	$150
2,500	$220
5,000	$343
10,000	$584

9" x 12" (4/0) POCKET FOLDERS	
500	$720
1,000	$864
2,500	$1301
5,000	$1824
10,000	$3071

8.5" x 11" (4/4) TRI-FOLD BROCHURES	
500	$175
1,000	$195
2,500	$261
5,000	$375
10,000	$601

2" x 3.5" (4/4) BUSINESS CARDS	
500	$ 12
1,000	$ 14
2,500	$ 45
5,000	$ 65
10,000	$104

Log on now for FREE Samples!

GREAT prices...*FAST* turnaround!

www.PrintDirectforLess.com

initially planned because what counts, first and foremost, is that it is fully updated and really ready to go. However, once version 1.0 has been launched, you need to know when version 1.1 will be ready for release. You can be sure of one thing: People are only truly passionate about software that is updated on a regular basis (and with which YOU earn money for a living). Just take a look at the sales chart of VoodooPad in our last article and you will clearly see how sales dipped down due to the lack of regular updates. You need to trust your estimation.

Let's say it always takes twice as long as your first estimate (this is your estimation factor): In the future, multiply your estimation with that factor and you will be much more precise. Believe me: You'll love the feeling of being able to implement a feature on time!

But new ideas pop up continually

Congratulations! That's quite normal and a sign that your creativity is in great shape, too.

When a new idea comes up while you're in working mode, jot your note down and continue to work. In this way, you stay focused and your mindset stays creative. It's not a vicious circle, it's a victorious circle – so take pride in it.

Don't think about your new idea for two or three days before you compare it with your existing ideas for version 1.0. In order to ensure adherence to the planned release date, you might need to exchange your new idea with an existing one.

But whenever in doubt, just stick to your original plan and save the new idea for a future version.

Decisions, Decisions, Decisions and one log

A decision log is simply a document listing all your decisions by topic, date of decision and description of the decision. Simple as it may seem during the development phase, it becomes super important later on and it's one of the very best time-saving tools.

You will have to take a lot of decisions right on up to the day when version 1.0 is on the shelves, and just waiting for buyers. In certain cases even after several months have passed, you'll find yourself confronted with a situation where you must make a decision on the very same topic again. Since you're smart, you don't waste any time by going through the whole decision-making process again. No, what you do is open your decision log up and quickly read about how and why(!) you decided months ago to take the course of action that you did. Now, all you need to do is to consider your decision in detail:

Do my reasons still hold true? If they do, it is okay and let's move on as planned.

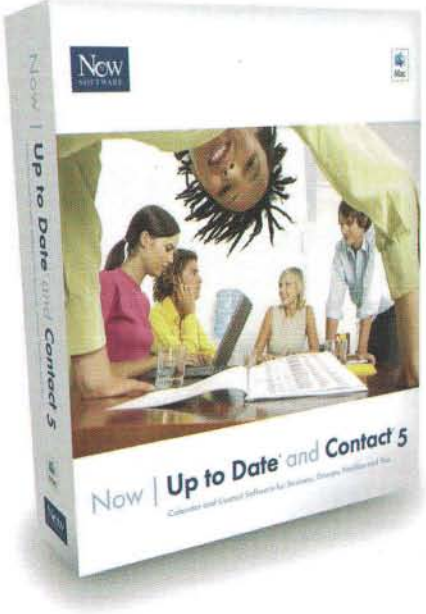
Are new facts influencing your former decision? Ok, let's take another, new decision. As a consequence, you might need to update your plan and document your decision in the decision log.

In most instances, former decisions still hold true and re-reading them will save you a great deal of time rather than re-deciding all over time and again.

Now **scheduling** and **contact management** for your entire organization.

Now | **Up to Date**® and **Contact**® **5**

Calendar and Contact Software for Business, Groups, Families and You.



Is this project on schedule? When are you available to meet about the systems upgrade? Where are all the field techs today? When was the last time anyone talked to our biggest customer?

Virtually all groups live (or die) by their abilities to meet deadlines and keep track of their customers, prospects, and vendors. Few small companies or even departments of big companies have the tools they need.

Now Up-to-Date & Contact might just be the calendar and contact software for you. It's time-tested and used by more Mac-based companies than any other solution. And it's cross-platform–available for your PC users, too. It's easy to install and manage and simple for your employees to understand and use.

Using Now Up-to-Date & Contact you can schedule meetings for multiple users, view multiple, simultaneous calendars, and reserve rooms and resources. You can share contact information about your customers, prospects and vendors. And using our free server software you can set it up in minutes and share with users in the office or from anywhere with an internet connection.

Call us now at 866-527-0556 or email us at mactech@nowsoftware.com and we'll send you our free evaluation kit, including the book that will make it all easy, "Take Control of Now Up-to-Date & Contact" from Take Control books!

NOW SOFTWARE

Phone: 866-527-0556

Web: www.nowsoftware.com

Feature	Date of decision	Decision
Outline View	03.10.2009	We only have time to implement an outline view or an attachment view. Implementing both would take too much time. We agree that an attachment view is more valuable to the majority of users than an outline view. The potential users we asked agreed on that. Yet, ideally they want both.
Outline View	05.05.2009	Since we have an additional developer we are now able to add the outline view as well.

Figure 6: Example of a decision log

To be honest, we did not use a decision log at the start and we often regret not having done so right from day one.

Figure 6 (above) is part of the decision log for Aphorism inspired by Bob Walsh.

Conclusion

During the planning session, you made up your mind about all of the features you need to implement for version 1.0. You now know what lies ahead. When you check your plan today, you might even come across certain dates where marking a time to celebrate would be absolutely fantastic!

The closer you get to the release date, the more you'll consider your plan to be like an assistant that helps you to stay in control and to not forget something that is really important (like the press release). The plan will help you to relax and it will sooth your nerves.

If you implement a feature and you have to make a critical decision, the plan and the decision log will support you in taking the best possible decision in terms of the context in which your feature will be implemented.

What's next?

Check your plan out: Are you or your team able to do it all by themselves? The GUI and Icon design, the coding, writing the user's manual, developing the webpage and crafting a press release that will spark potential buyers' interest?

Not all of us have been blessed with the gift of an omnipresent talent like that (I'm not). In our next article, we will tell you how to find the "Seven Samurai" who will help make your dreams come true.

If you're curious and want to find out more about the above-mentioned topic now, we highly recommend checking the "Seven Samurai" DVD out by Akira Kurosawa. The plot is the prototype of all modern action films and it's the metaphor of our next article.

Connect with us!

We want to share stimulating, innovative ideas with you and we really look forward to your feedback! Is anything missing or do you think something could be fleshed out in further detail? Just let us know and write to oliver.pospisil@inspiredbylife.com.

Bibliography and References

Books:

Snyder, Carolyn. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. San Diego, 2003.

Johnson, Spencer. *Yes or No: The Guide to Better Decisions: A Story*. New York, 1992.

Dan Roam. *The Back of the Napkin*. New York, 2008.

Bob Walsh. *Micro-ISV: From Vision to Reality*. New York, 2006.

Websites:

iPhone stencil:
http://www.designcommission.com/shop/iphone-stencil-kit/

Joel Spolsky on feature spec (Part 1 of 4):
http://www.joelonsoftware.com/articles/fog0000000036.html

Joel Spolsky on estimation:
http://www.joelonsoftware.com/articles/fog0000000245.html

Cultured Code – Things for iPhone:
http://culturedcode.com/things/iphone/

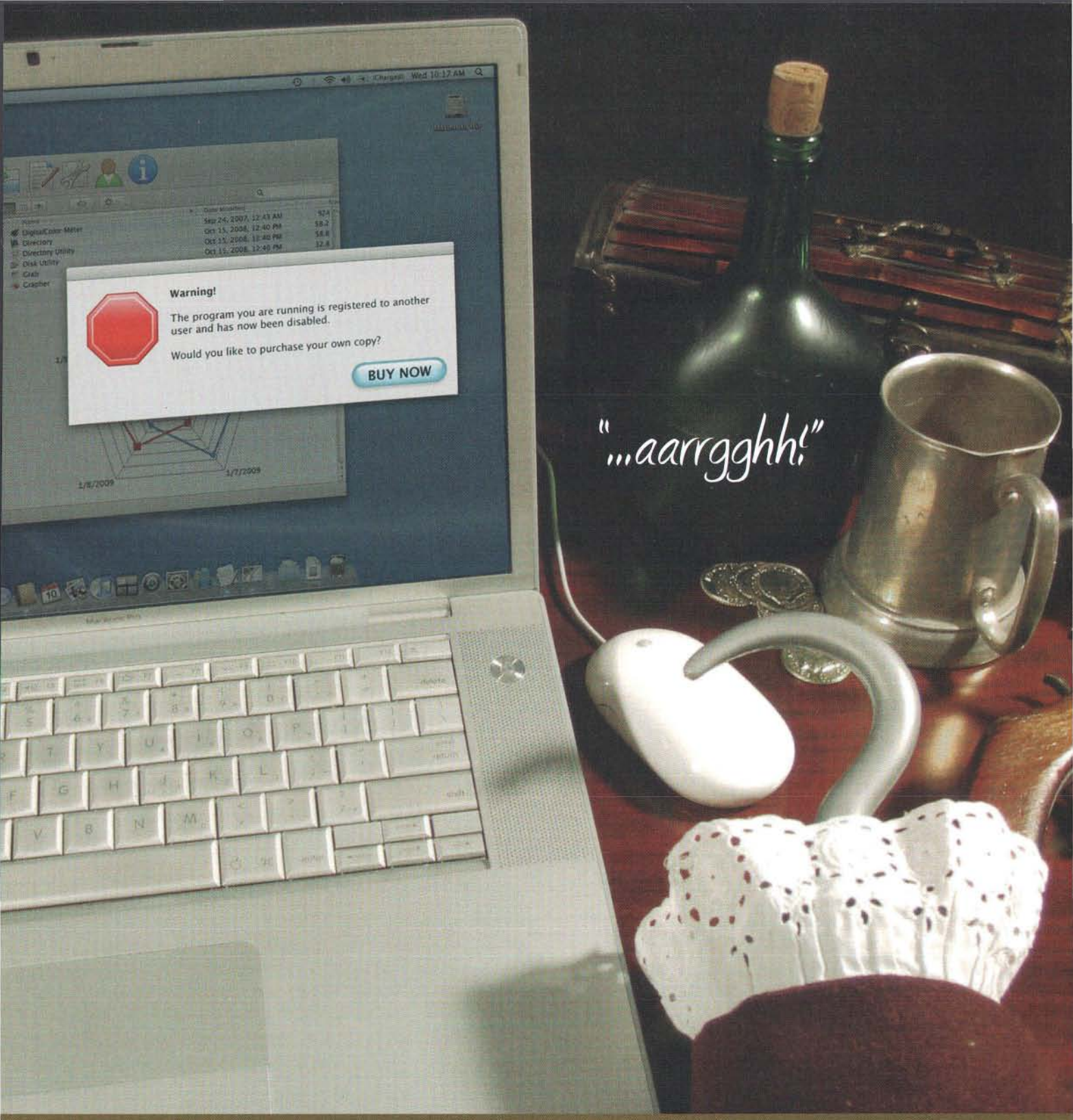
Merlin: http://www.projectwizards.net/en/merlin/

FogBugz: http://www.fogcreek.com/FogBugz/

About The Authors

Michael started MOApp up in 2004 and he has now developed more than ten applications - six of them are Apple staff picks. He does everything from software development, icon design, website development to sales management and public relations.

Oliver has been in the software business for over ten years, specializing in areas ranging from Palm programming to large-scale, in-house Java projects. In 2006, an idea grabbed his attention that both are now working on. He is still working full-time for a German retail company and will be until the new business starts paying off their bills.



eSellerate Product Activation.

Protectin' yer booty from scurvy dogs.

SELL. MANAGE. PROTECT.

www.esellerate.net/aarrgghh

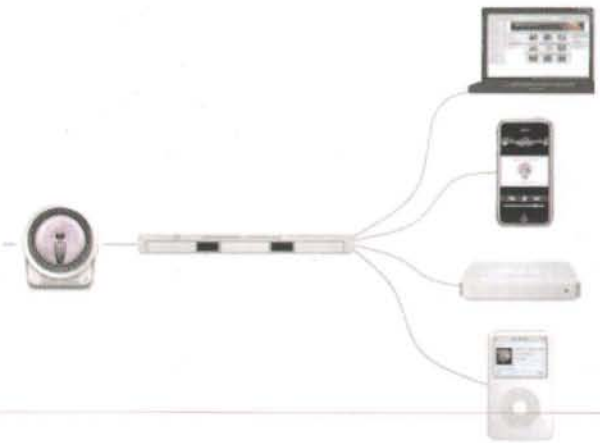
Unlocking Podcast Producer v2

by Michele (Mike) Hjörleifsson

Part One: What is Podcast Producer?

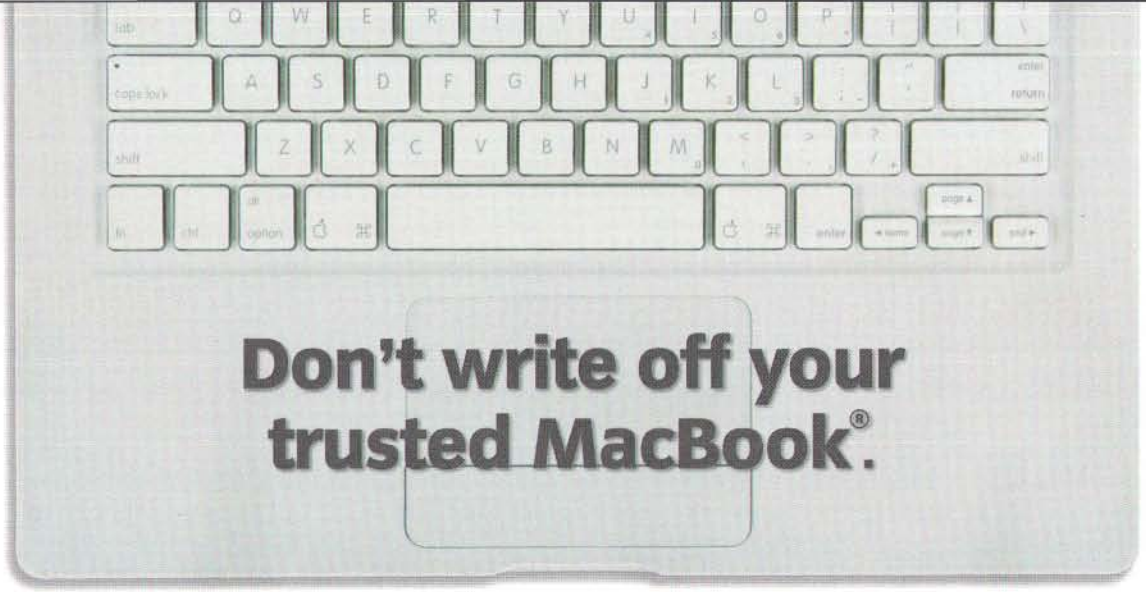
A podcast by any other name like vidcast, screencast, audiocast, doc-cast, and so on is still a podcast; well, at least for the context of this series of articles and from Snow Leopard server's point of view. Podcast producer, simply put, does for podcast production and distribution what the bread maker has done for homemakers around the globe—simplified the tedium so the cook can focus on the content. This series of articles introduces you to Podcast Producer, its history, potential uses, and strengths, and includes some practical use scenarios and how-to's.

Processing podcasts for consumption can be a tedious process from the initial content creation to the transcoding for desktop, iPod, iPhone, audio only, HD and transcribed formats. Delivery is no small task either, posting to blogs, creating RSS/Atom, sending to a digital archive, posting to iTunes, posting to a website or Sharepoint site and so on more tedium. As you can see in the following figure there are a lot of wheels that need to turn to grind that wheat into usable flour.

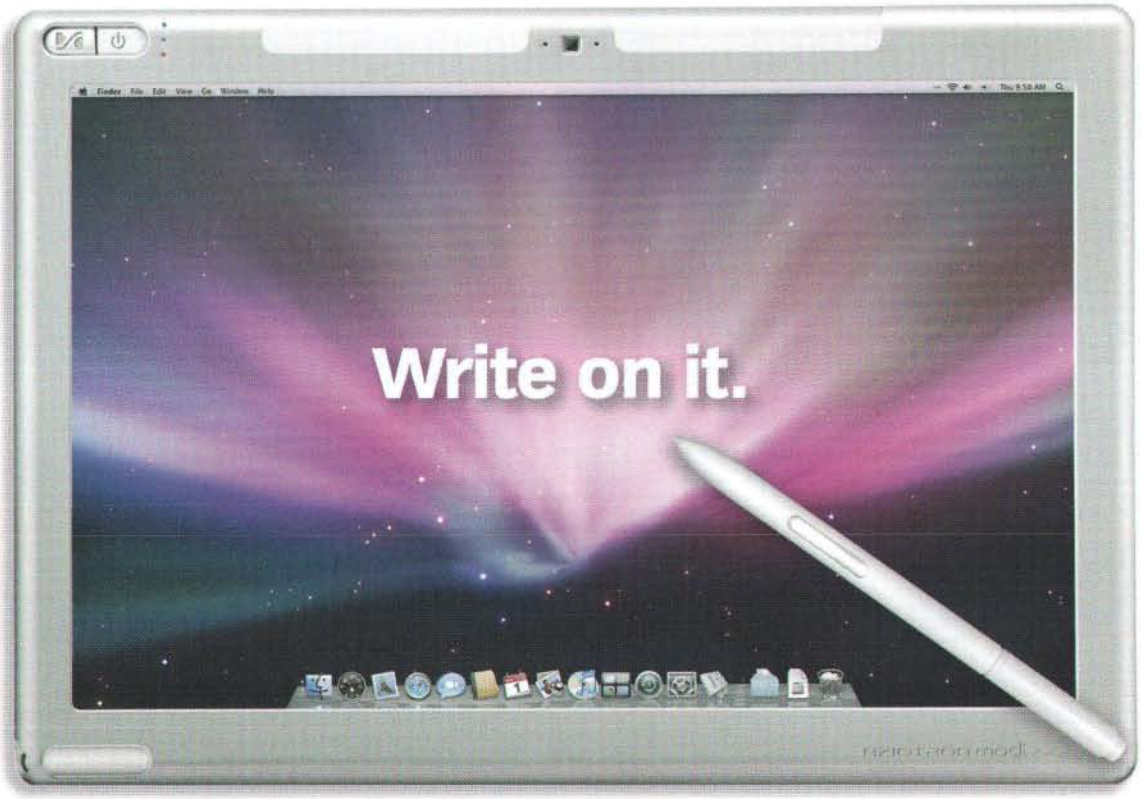


Podcast Producer version 1 is part of Leopard server. Like XGrid and WebObjects, it is an under-discussed, under-promoted useful (and cool) piece of technology. Using Podcast Producer in Leopard Server, administrators work with the content producers to create a recipe, called a workflow, which is a bundle of XML files, associated media and resources that determine the processing of the content, the notification facilities to be used, archival (if necessary) and distribution to a wiki/blog, QuickTime Streaming or iTunes. Once created, the same workflow provides an automated engine to process and distribute content into a podcast. How wonderful— a collection of software and services that do what computers were originally designed to do: remove repetitive tedious tasks. Well, maybe not so wonderful. As anyone who deployed Podcast Producer version 1 will tell you, it takes a bit of knowledge of Kerberos, Xgrid and XML to get the service humming. But, once in place it is a wonderful toolbox. The most tedious part of the getting your Podcast Producer infrastructure going is editing the XML files that outline your recipe (the processes and dependencies applied to your content). Though, not much fun but there is a bright light on the horizon. Snow Leopard has a new version of Podcast Producer with some awesome tools and capabilities. Let's explore.

Podcast Producer version 2 is also a free service provided in Snow Leopard server and it has seen some significant enhancements. First, there is an assistant to establish the required configuration of the underlying services like Xgrid (and NFS to facilitate shared resources), Kerberos and the Podcast Producer service itself. Not to be left behind is the new Podcast Composer, my personal favorite. This Automator-like tool provides a graphical, intuitive way to create your podcast recipe. No more of those pesky XML files, at least not yet. But wait, there's more! (A little tribute to the late Billy Mays). The client software called Podcast Capture (which is included, you guessed it, free of charge, on every Snow Leopard client machine) can now perform simultaneous capture of screen and video with an iChat Theatre look and feel—polished and professional. There is an official web capture client to allow non-OS X users to become content producers as well (there was an



Don't write off your trusted MacBook®.



Introducing Axiotron® Modservice™ — transform your *existing* Apple® MacBook into an Axiotron Modbook®.

It works like this: Sign up for Modservice.* Choose your upgrade options and the warranty you want. Turn in your computer to one of our Axiotron Authorized Service Providers for conversion. Then kick back and enjoy your new Modbook. Draw, sketch and write by putting a pen to the screen of the best tablet computer with industry-leading Wacom® Penabled® technology and Mac OS® X. Portable and versatile, the Modbook empowers your creativity and imagination.

With Axiotron Modservice, your Apple MacBook computer gets revitalized, renewed and revolutionized — into a whole new product, the award-winning Axiotron Modbook. Visit www.axiotron.com/modservice for details.

the one and only AXIOTRON modbook

*Some service limitations and restrictions apply.

unsupported web interface to the previous version that came, went, then came back). Newly baked into Podcast Producer are some integration bits with Final Cut Server, which provide things like approval and review workflows and the power of Final Cut Server's new metadata processing and automated tasks. Last, but certainly not least, (especially to the MacTech community) is the ability to call out to command line tools of your choosing so you can reach out and extend Podcast Producer to limitless potential scenarios. Included under the hood is a set of great command line tools to do some of the basics and more advanced functions like calling and creating Quartz Composer animations. Oh wait, did I mention that QuickTime Streaming server also got an overhaul? Although this is not directly related to Podcast Producer, I must tell you about the ability to produce a podcast that you can then stream to your iPhone, yes STREAM, not sync to your iPhone.

So what can we use this cool set of tools for? Ahh, glad you asked. First there are the traditional podcasts, like recording educational lectures at K-12 schools and universities for students, the general public, or both (there are hundreds of these for free download on iTunesU). That was a pretty obvious example but let's take a deeper look.

Problem: The economy is tight, you need to get corporate messaging and training out to your ten, hundred or thousand remote locations or retail presences.

Solution: Produce a podcast with your existing video conferencing gear (more on this later), a Mac or PC and send it to Podcast Producer for distribution to Apple TV units hooked to television at your remote locations.

Problem: You need to provide regulatory, mandated continuing education to your legal, medical or financial industry employees.

Solution: Develop a workflow that provides random codes as a watermark in your podcasts to ensure staff members actually watched the entire podcast.

Problem: You have a new national product launch and you need to get your field sales folks trained on the features, messaging and look-and-feel of the new gear.

Solution: Develop a workflow that encodes your podcast for streaming to their iPod Touch or iPhone devices and send them an email notification when it is ready.

Problem: You need to increase readership or web presence for your magazine newspaper, television or radio station.

Solution: Add deeper look podcasts that takes a corresponding article, episode or segment to the next level with multimedia (for a great example. visit http://www.apple.com/pro/profiles/washingtonpost/). Send the podcast to Final Cut Server for editor and legal review and approval then Podcast Producer will slice and dice your content to a video iPod version, video iPhone version, Desktop Video version, and an audio

Macs Need Doctors Too.

PowerbookMedic

Your Mac. Our Patient.

Powerbook Medic is the ultimate resource for all your Mac needs.

Browse our online catalog for hundreds of repair parts. Being more than just a store, we offer **free** repair guides on all popular Apple products so you can learn how to do the repairs yourself, or you can take advantage of our send-in service and let us fix it.

The most affordable. The most knowledgeable. The best customer service.

CONTACT US TODAY

PowerbookMedic.com

1-866-726-3342

only version for your Apple Shuffle. Then, consumers post it to your website and update RSS feeds to inform the subscribers.

Problem: You are an expert in a certain field and would like to provide "paid for content" to the public but you don't know a thing about processing media files for consumer production.

Solution: Work with an Apple Consultant to setup a Podcast Producer server to process your media and post it to a high end video management system like Status Firm's Core Nucleus, or freeware blogs with pay for content plug-ins such as Joomla or Drupal (we will explore an example with Joomla in an upcoming segment).

Problem: You are recording video testimonials about a terrible situation somewhere that may later be used in a court trial. How do you ensure the validity of the videos three, five or ten years from now so it's admissible as testimony by government standards?

Solution: Create a workflow that calls a script you created that digitally signs the video with a PKI certificate you have taken appropriate precautions to secure. (Refer to the set of articles on PKI that I authored for MacTech on how to generate these types of certificates.)

So there are six quick examples to get you thinking. The point here is that anytime you need to get information in the hands of folks that is better presented in multimedia format you will need to produce that content and deliver that content and that is where Podcast Producer comes in.

The three top features of Podcast Producer have nothing to do with the cool media processing it can do, at least not from your point of view as a technical Apple person.

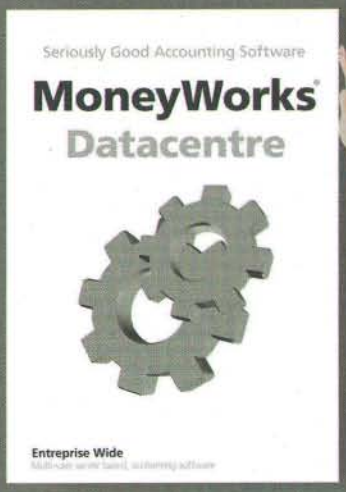
Scalability: Utilizing Xgrid you can process thousands of pieces of inbound media into a plethora of formats in a predictable amount of time.

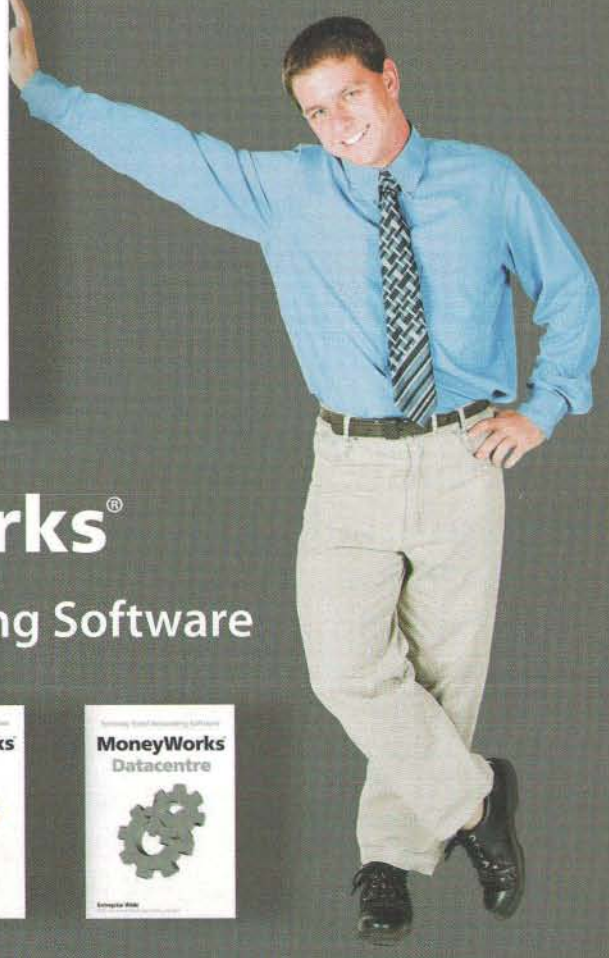
Extensibility: Podcast Composer allows you to call your own scripts, which allows you to literally do anything your powerful cerebral processor can crank out.

Interoperability: The web services provided by Podcast Producer allow you to utilize existing video conferencing gear like Tandberg's TCS content server to record content from any standards compliant video conferencing gear and ingest that media to your workflow. The web services also allow you to support non-Apple client devices to create and upload content, or even create your own web or even iPhone application to get content into Podcast Producer to slice, dice and deliver. Apple has extended Podcast Producer with the new web interface and integrated Final Cut Server allowing you to ingest content from the web and post content to Final Cut Server for processing, and approval workflows. Last but not least, interoperability allows you to take content and process it for delivery via QuickTime to your iPhone users.

So now that your head is spinning and gears are turning on all the things you would like to try out, let's get cracking.

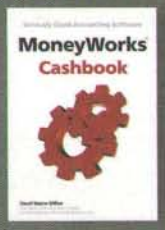
Sometimes the Tech Guy makes the best choices for the company





MoneyWorks®

Seriously Good Accounting Software



For all sizes of Mac and/or PC based businesses

Learn more at www.moneyworks.ca/us

Info@moneyworks.ca or call 800•818•6955

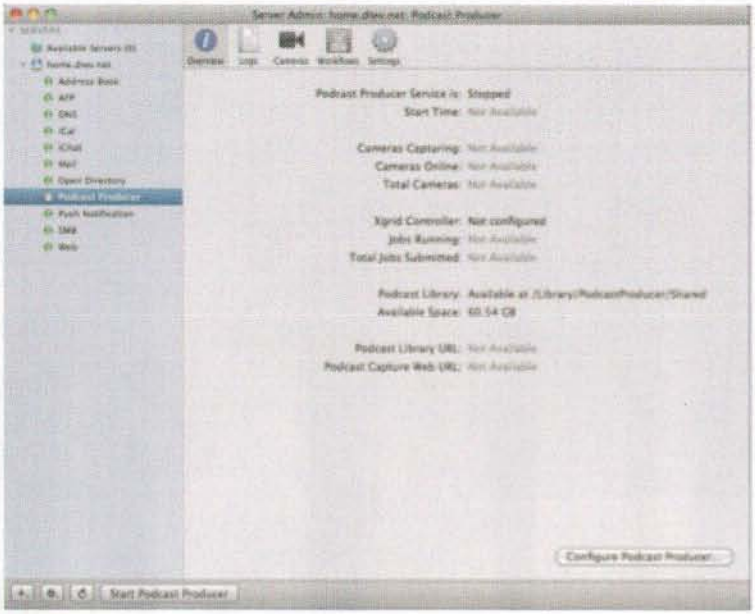


Figure 2

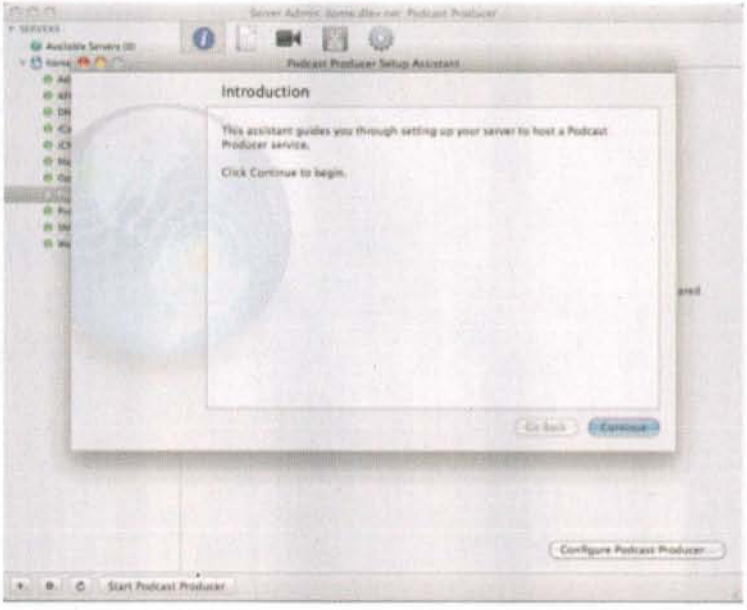


Figure 3

Figure 4

First, on your test Snow Leopard server make sure you have a standard install and created a test user account to play with. Click the servers name in *Server Admin* then click the *Services* tab. Select Podcast Producer and then click *Save (see Figure 2)*.

This does NOT enable Podcast Producer; rather, it provides you with the settings panels to set up and then enable Podcast Producer. Click on the new Podcast Producer item on the left side of *Server Admin*. Then, click the *Configure Podcast Producer* button to get started *(see Figure 3)*.

Accept the defaults and voila, you have a working Podcast Producer. For those of you who configured version 1 of Podcast Producer you should be basking in the ease of the new tools. Now let's go to a Snow Leopard client. Open the Applications->Utilities folder and then open Podcast Capture. Log on to your Podcast Producer server. Click Audio and then submit to one of the default workflows *(see Figure 4)*.

Now let's take a look at the web client. Open Safari on your client machine and then connect to your server on port 8188 (i.e. http://myserver.local:8188), You can log on here and submit content to the same default workflows.

That's all there is to it. You are now up and running. I urge you to play with Podcast Composer on your server and see the plethora of options there are for creating workflows.

Next month we will explore the following: default workflows and how to secure them by setting permissions.; customizing the workflows with external scripts to really harness the power of Podcast Producer; extending Podcast Producer to provide content to Joomla. Thank you for spending the time with learning about the Podcast Producer. I hope you explore the technology and have as much fun with it as I have had.

MT

About The Author

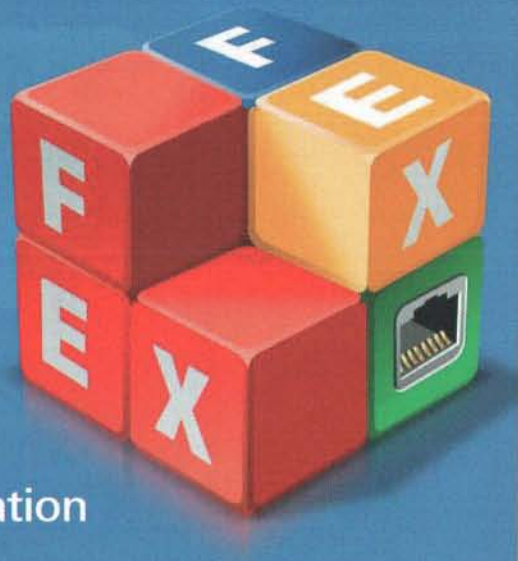
Michele (Mike) Hjörleifsson has been programming Apple computers since the Apple][+, and implementing network and remote access security technologies since the early '90s. He has worked with the nation's largest corporations and government institutions. Mike is currently a certified Apple trainer and independent consultant. Feel free to contact him at mhjorleifsson@me.com

ESC> the pain of font management

ESC> frequent font server crashes

ESC> font licensing headaches

ESC> limited font selection for creatives



ESC> unintuitive font management administration

ESC> the tech support runaround

ESC> multiple contacts for font management, font purchases, font license management and other font needs

ESC> the pain of font management with the FontExplorer® X Server solution

FontExplorer X: font management from the font experts

Linotype and Monotype Imaging have teamed to bring you FontExplorer X Server. This unmatched font management solution combines a robust feature set that the creative workflow demands, a solid architecture and an intuitive nature that network administrators need, and a selection of world class fonts that creatives crave. With powerful font management capabilities and licensing for our renowned Linotype®, Monotype® and ITC® OpenType® font libraries, FontExplorer X Server provides an unprecedented solution for managing fonts across creative organizations or large design groups. See how your organization can benefit from centralized font management — and the convenience of having a single contact for font management, font licensing, font license management and any other typographic request.

Call us at (800) 424-8973 for a free evaluation.

Monotype Imaging

Monotype Imaging Inc.
500 Unicorn Park Drive
Woburn, MA 01801
Toll Free: (800) 424-8973
Phone: 781 970 6000

Linotype

Linotype GmbH
Werner-Reimers-Straße 2-4
61352 Bad Homburg
Germany
Phone: +49 (0) 6172 484-418

www.fontexplorerx.com

Snow Leopard, launchd, and Lunch

More launchd recipes, and a look at changes in Snow Leopard

By Greg Neagle, MacEnterprise.org

MacEnterprise.org
Mac OS X enterprise deployment project

Introduction

In a prior column, we looked at some "recipes" for using launchd for systems administration tasks. The recipes showed how to use launchd to run a script at system startup, on a repeating schedule, and when a filesystem item changed. Finally, we explored a way to use launchd to allow a non-admin to run a script with root permissions.

I promised a few more launchd recipes. We'll get to those, but first, with the release of Snow Leopard, let's look at some of the changes to launchd in OS X 10.6.

New in Snow Leopard

There were several major changes in launchd from 10.4 to 10.5. Among these were the more powerful `KeepAlive` options, and the new `LimitLoadToSessionType` key, which enabled LaunchAgents to load only in specific contexts, like at the loginwindow, or only during SSH logins. The changes in 10.6 are a bit more subtle, and some of the changes are probably of more interest to software developers than systems administrators.

Launchd plists gain a new optional key: `EnableTransactions`, which is a Boolean value. If this is set to true, it means the daemon will use the vproc_transaction_begin and `vproc_transaction_end` system calls to mark outstanding transactions that must be handled before the daemon can be safely terminated. This is an element of the new "faster shutdown" feature of Snow Leopard – if there are no outstanding transactions, launchd will send a SIGKILL signal to the process instead of SIGTERM.

Launchd contexts are now unified between GUI logins and command-line logins: for example, you can copy some text in TextEdit, then ssh into the machine as the same user and do a `pbpaste`, retrieving the copied text. This wasn't possible before.

There are more restrictions on what root can do to a user session. Apple strongly recommends that if you want a process to run as a specific user, with that user's environment, that you should make the process a launchd agent. Simply performing a `setuid` or calling `su username` to "become" a user is no longer recommended or officially supported.

The Disabled key

The biggest change in Snow Leopard of interest to systems administrators is how launchd and launchctl handle the `Disabled` key in the launchd plist. Prior to 10.6, if you disabled a launchd job using launchctl like this:

```
launchctl unload -w /path/to/launchd.plist
```

The job would be unloaded, and the `Disabled` key in the launchd plist would be set to `true`, causing the job to be disabled. In Snow Leopard, the job is still marked as disabled, but the plist is not changed. The value of the `Disabled` key is stored elsewhere. The `launchctl` man page is vague about where it is stored, but it turns out to be in `/private/var/db/launchd.db/`.

Inside this directory, there are subdirectories like these:

```
aquaman:launchd.db root# ls -1
com.apple.launchd
com.apple.launchd.peruser.0
com.apple.launchd.peruser.100
com.apple.launchd.peruser.212
com.apple.launchd.peruser.501
com.apple.launchd.peruser.97
```

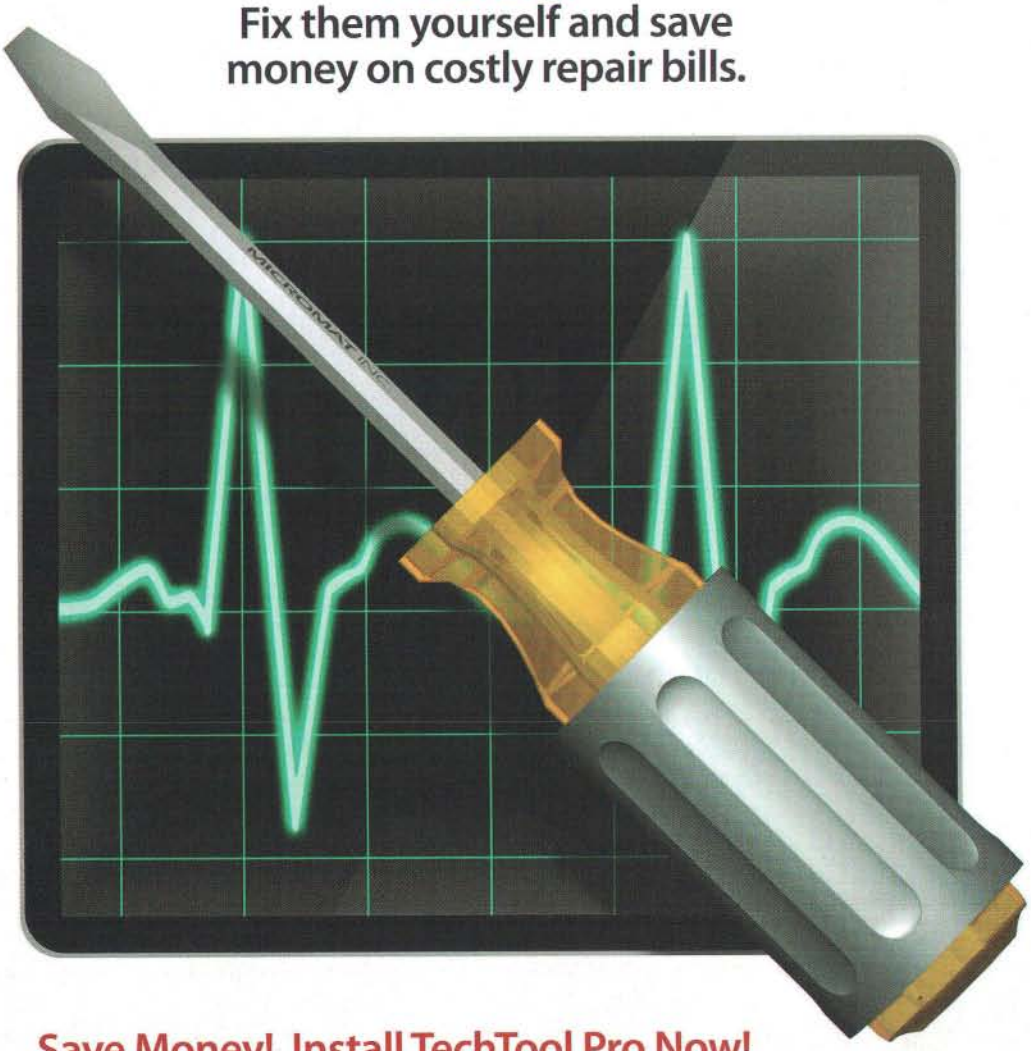
The `com.apple.launchd` directory holds info for LaunchDaemons, the `com.apple.launchd.peruser.*` directories hold info for LaunchAgents. Let's look a little deeper:

```
aquaman:launchd.db root# cd com.apple.launchd
aquaman:com.apple.launchd root# ls
overrides.plist
```

Let's examine the contents of `overrides.plist`:

TECHTOOL PRO 5

Problems?
Fix them yourself and save money on costly repair bills.



Save Money! Install TechTool Pro Now!

- Keep your computer running smoothly.
- Help prevent problems in the future.
- Fix most of your problems on the spot.

Mac Universal

Micromat, Inc., 5329 Skylane Blvd., Santa Rosa, CA 95403, USA
800-829-6227 707-566-3831 info@micromat.com www.micromat.com

AcademicSuperstore.com campustech.com frys.com drbott.com bestbuy.com navarre.com macmall.com store.apple.com microcenterorder.com

```
aquaman:com.apple.launchd root# cat overrides.plist
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>com.apple.backupd-attach</key>
    <dict>
        <key>Disabled</key>
        <true/>
    </dict>
    <key>com.apple.backupd-auto</key>
    <dict>
        <key>Disabled</key>
        <true/>
    </dict>
```

...and so on. So we see that the current state of the `Disabled` keys for LaunchDaemons is stored in `/var/db/launchd.db/com.apple.launchd/overrides.plist`.

Since each user now has a separate directory under `/var/db/launchd.db/`, LaunchAgents can now be enabled/disabled on a per-user basis. In other words, a LaunchAgent located in `/System/Library/LaunchAgents/` or `/Library/LaunchAgents/` can now be disabled for a single user. For example, I could disable the WacomTabletDriver for only my login:

```
[aquaman:~] gneagle% launchctl unload -w -S Aqua
/Library/LaunchAgents/com.wacom.wacomtablet.plist
[aquaman:~] gneagle% cd
/var/db/launchd.db/com.apple.launchd.peruser.4389/
[aquaman:~] gneagle% cat overrides.plist
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>com.wacom.wacomtablet</key>
    <dict>
        <key>Disabled</key>
        <true/>
    </dict>
</dict>
</plist>
```

The result is that when I log in, the WacomTabletDriver process does not load, but if any other user logs in, the process will load. In prior versions of OS X, there was no way to enable/disable launchd jobs on a per-user basis – enabling or disabling a LaunchAgent in `/System/Library/LaunchAgents` or `/Library/LaunchAgents` affected all users of the machine. On the other hand, it is now possible for non-admins to turn off LaunchAgents that run in their context. This might be a problem if you rely on LaunchAgents to run at login and perform certain tasks for the user – the user can now turn these off.

This change may also make it more difficult for a systems administrator to determine the effective enabled/disabled state of a LaunchAgent; the administrator must first check the launchd job's plist to get the initial state of the Disabled key, then check `/var/db/launchd.db` for any overrides.

Finally, there is a new ServiceManagement framework to provide a supported API to get a list of launchd jobs, submit new jobs to launchd, and to securely install privileged helper tools. Documentation is scarce as of this writing; if you have Snow Leopard and Xcode installed, you can see some basic info in the ServiceManagement header file located at:

```
/System/Library/Frameworks/ServiceManagement.framework/Headers/ServiceManagement.h
```

Traditionally, framework APIs like this were mostly of interest to C coders and application developers. But with OS X's BridgeSupport, which allows Python and Ruby access to many OS X frameworks, Python and Ruby scripters can make use of these APIs.

That's a look at the launchd changes in Snow Leopard. Now let's get back to launchd recipes!

Recipe 5: Run a script (or an application) when a user logs in

This recipe makes use of a launchd LaunchAgent. By default, these are loaded when a user logs into a GUI session, and run as the user. There are several third-party software items that install LaunchAgents to run a background process when a user logs in. Some examples include the Wacom Tablet software, which launches a driver process for the tablet at login, and Timbuktu, which loads the Timbuktu Host.app at login.

You can leverage this same technique to run your own scripts. Perhaps you've written a Setup Assistant for your organization that helps your users configure their mail accounts and so on when they first login. You'd like that assistant to launch automatically at login the first time the user logs in.

Ingredient 1:

Your setup assistant application. For this recipe, it will be `/Application/Utilities/MyOrg Setup Assistant.app`. If you don't have a setup assistant application, but you do have a web page of setup instructions you'd like your users to follow, you can replace the path to the setup assistant app with an http URL like so: "http://www.myorg.com/help/firsttimeusers/"

Ingredient 2:

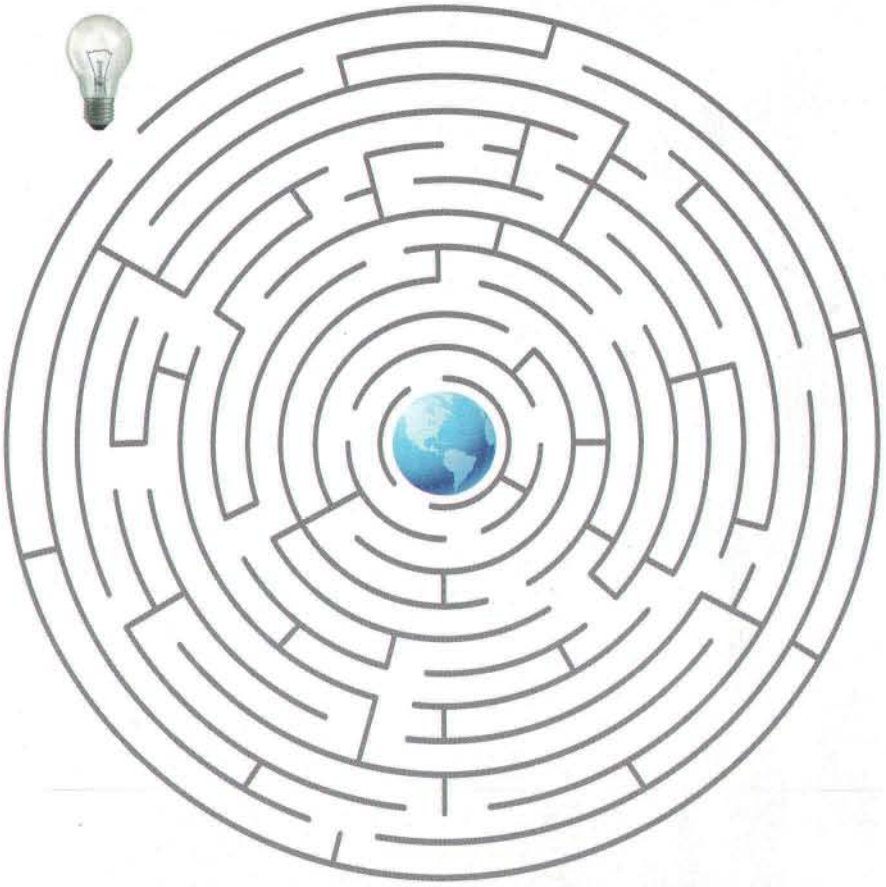
A shell script that checks for the existence of a file, say `~/.com.myorg.setupassistant.done` and if it doesn't exist, creates the file, then launches your assistant (or opens your web page). Here's an example script:

```
#!/bin/sh

FLAGFILE=".com.myorg.setupassistant.done"
SETUPASST="/Applications/Utilities/MyOrg Setup
Assistant.app"
cd ~
if [ ! -f "$FLAGFILE" ] ; then
    touch "$FLAGFILE"
    open "$SETUPASST"
fi
```

Access a worldwide market

We'll lead your product through the maze



Reach a global retail audience with Smith Micro Software.

Working on the next great Mac Application? Love to see your App on store shelves and catalog pages but you don't know where to start? Get into stores with Smith Micro. We have over 20 years experience publishing and distributing award-winning products for Mac and Windows markets. We do it all—package design, marketing and manufacturing, so you don't have to.

Great products published by Smith Micro:

smithmicro
software

Talk to us to get started:
email **bizdev@smithmicro.com**

This is basic shell scripting, nothing fancy here. We define a couple of variables, change to the current user's home directory, then check the existence of the flag file.

If the flag file is missing, we touch the flag file, which creates an empty file of that name, then we open the setup assistant.

We name the script "run_setupassistant.sh", save it someplace convenient, and make sure to mark it as executable:

```
chmod 755 /path/to/run_setupassistant.sh
```

Ingredient 3:

A plist to get launchd to run the script at login. Here's one:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST
1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>Label</key>
    <string>com.myorg.setupassistant</string>
    <key>Program</key>
    <string>/path/to/run_setupassistant.sh</string>
    <key>RunAtLoad</key>
    <true/>
</dict>
</plist>
```

We'll save this plist as `/Library/LaunchAgents/com.myorg.setupassistant.plist`, and make sure the owner and permissions are correct:

```
cd /Library/LaunchAgents
sudo chown root:wheel com.myorg.setupassistant.plist
sudo chmod 644 com.myorg.setupassistant.plist
```

Once all the ingredients are in place, we can test by logging out and back in. If you did everything right, your application will launch, or your web page will open in your default browser. Log out and back in again to verify that the assistant doesn't launch on subsequent logins. Remove `~/.com.myorg.setupassistant.done` (or whatever you've named your flag file) and log out and back in again, and your assistant should launch again.

(If you want to play with this example, but don't have a custom Setup Assistant handy, just change the script to open any application or URL you want.)

Program vs. ProgramArguments

If you read my previous column on this subject, and you have a photographic memory, you might have noticed that I'm specifying the program to be run differently in this plist. In all of the prior recipes, I used the `ProgramArguments` key, like this:

```
<key>ProgramArguments</key>
<array>
  <string>/usr/sbin/softwareupdate</string>
  <string>-install</string>
     <string>-all</string>
</array>
```

Email Marketing Made Easy... Just Like the Mac!

The standard in permission-based email marketing.

- Grow your contact lists
- Send quality emails
- Track your responses
- Survey your customers

Plans starting at only $9.99 per month

Sign up for a FREE 30 Day Trial Today!

www.BenchmarkEmail.com

benchmark email

800.430.4095

but in this recipe, I'm using the `Program` key:

```
<key>Program</key>
<string>/path/to/run_setupassistant.sh</string>
```

The `Program` key takes a single string; `ProgramArguments` takes an array of strings. You can use the `Program` key if you are specifying the path to an executable that needs no arguments passed to it. Otherwise, you should use the `ProgramArguments` key, with the first string in the array being the path to the executable, and subsequent strings each containing one argument. It's possible to use both keys in a plist, but since you'd have to specify the executable path in both keys, it's pretty pointless. For my own use, I tend to stick with `ProgramArguments`, even if there are no arguments.

Recipe 6: Run a script at the loginwindow

OS X 10.5 introduced the ability for launchd to run a job at the loginwindow. To do this, you need a special type of LaunchAgent.

Ingredient 1: The script.

For this recipe, we're going to change the image behind the loginwindow to a randomly selected image at each login. Here's the script:

```
#!/usr/bin/perl -w
use strict;

my $loginwindowprefs =
"/Library/Preferences/com.apple.loginwindow";
my $picdir = "/Library/Desktop Pictures/Nature";

if ( -d "$picdir") {
    my @list = split("\n",`ls -1 "$picdir"`);
    my @pictures = ();

    for my $item (@list) {
        if (-f "$picdir/$item") {
            push @pictures, "$picdir/$item";
        }
    }

    if (scalar(@pictures)) {
        my $currentpicture = `/usr/bin/defaults read
$loginwindowprefs DesktopPicture`;
        if ($currentpicture) { chomp($currentpicture) };
        my $randompicture = $currentpicture;

        while ($randompicture eq $currentpicture) {
            my $randomindex = int(rand(scalar(@pictures)));
            $randompicture = $pictures[$randomindex];
        }

        my $result = `/usr/bin/defaults write
$loginwindowprefs DesktopPicture "$randompicture"`;
    }
}
```

Sorry about the Perl! I wrote this years ago – it would probably be more readable written in Python, but it gets the job done. Here's what the script does:

Gets a list of all the pictures in `$picdir` and puts them into the `@pictures` array.

Gets the path to the current DesktopPicture (the one behind the loginwindow) by calling `defaults read /Library/Preferences/com.apple.loginwindow DesktopPicture`

Picks a picture at random from the `@pictures` array.

If the picture chosen is the same as the current one, try again until we pick one that's different.

Set the loginwindow background by calling `defaults write /Library/Preferences/com.apple.loginwindow DesktopPicture /path/to/new/picture`

Save the script as `/Library/Scripts/loginwindowPictureChanger`, and make sure it's executable.

Ingredient 2: The plist.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Label</key>
  <string>com.myorg.loginwindowpictures</string>
  <key>LimitLoadToSessionType</key>
  <array>
    <string>LoginWindow</string>
  </array>
  <key>Program</key>
<string>/Library/Scripts/loginwindowPictureChanger</string>
  <key>RunAtLoad</key>
  <true/>
</dict>
</plist>
```

The new bit here is the `LimitLoadToSessionType` key. By setting this to `LoginWindow`, launchd loads the job when the loginwindow displays. We also set `RunAtLoad` key to true so the script runs immediately when the job is loaded.

Save the plist as `/Library/LaunchAgents/com.myorg.loginwindowpictures.plist`, ensuring owner, group, and permissions are correct. See the previous recipe if you forgot how.

The script will now run each time the loginwindow displays.

If you try this, don't be surprised if it doesn't seem to work the first time you log out. Since the loginwindow is already loaded when our script runs, changing the value of the DesktopPicture in the defaults has no effect until the next time the loginwindow is loaded. So we're really changing the DesktopPicture for the *next* time the loginwindow displays, not the current time.

Recipe 7: Run a script when a volume is mounted

Prior to 10.5, the only way you could use launchd to run a script when a volume was mounted was to define a WatchPath

of "/Volumes". Your job would be launched on any change to /Volumes, which included volumes mounting and unmounting. And if a file system got mounted anywhere else other than /Volumes (not common, but possible), launchd would miss it. 10.5 added a new StartOnMount key which takes a Boolean value. This makes it much simpler to define a job that runs whenever a filesystem is mounted.

Possible uses for this include scanning for viruses on newly mounted filesystems, or doing an automatic backup of a specific directory anytime a specific FireWire or USB disk is mounted. I'll leave that part up to you.

Here's an example plist that runs /path/to/my_diskmount_script.sh whenever a filesystem is mounted.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST
1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>Label</key>
    <string>com.myorg.diskmountscript</string>
    <key>Program</key>
    <string>/path/to/my_diskmount_script.sh</string>
    <key>StartOnMount</key>
    <true/>
</dict>
</plist>
```

If you wanted it to run as root (for example, if was a virus scanner), you could make it a LaunchDaemon and put the plist in /Library/LaunchDaemons. If you wanted it to run as a user, you'd make it a LaunchAgent by putting it in /Library/LaunchAgents. If you implement the idea of an automatic backup script, you might want it to run only for a specific user (like you). In that case, you'd put the plist in your home directory:

```
~/Library/LaunchAgents/com.myorg.diskmountscript.plist
```

– which would cause it to load only when you logged in.

Conclusion

In two columns, we've looked at seven launchd recipes for common systems administrations tasks. We haven't exhausted every way a systems administrator could use launchd, but we have covered most of the common uses.

If you want even more info on some of the things we've covered here, check out these resources:

Apple Technical Note TN2083: Daemons and Agents
http://developer.apple.com/technotes/tn2005/tn2083.html
An in-depth tech note describing various techniques and issues when working with system daemons and user agents.

MacResearch: Tutorial: Backups with Launchd
http://www.macresearch.org/tutorial_backups_with_launchd
An example of using launchd to run an automatic backup when a disk in mounted. Written with 10.4's launchd in mind, but still contains useful information.

Launchd at Mac OS Forge:
http://launchd.macosforge.org/
http://launchd.macosforge.org/trac/wiki

Apple has released the launchd source code as open source, and it's available at Mac OS Forge. The site hasn't been updated in a while, but there are still some helpful tidbits and info from the launchd developers.

launchd-dev mailing list:
http://lists.macosforge.org/mailman/listinfo/launchd-dev

This list is supposed to be for the discussion of the development of launchd, but there are often "launchd-user" discussions on this list as well.

About The Author

Greg Neagle is a member of the steering committee of the Mac OS X Enterprise Project (macenterprise.org) and is a senior systems engineer at a large animation studio. Greg has been working with the Mac since 1984, and with OS X since its release. He can be reached at gregneagle@mac.com.

AppleScript in Snow Leopard

by Ben Waldie

With the release of Snow Leopard, AppleScript developers will no doubt be in store for a few surprises. While the AppleScript language itself hasn't changed too much, a variety of other enhancements are welcome, but will take some getting used to.

Noticeable Changes

Script Editor

One of the first things you will notice is that Script Editor is no longer found in the */Applications/AppleScript* folder on your Mac. Don't worry, though, it hasn't gone away. Apple has just moved it into a new, perhaps more appropriate, location – into the */Applications/Utilities* folder. In addition, to help better clarify its purpose to those new to AppleScript, Script Editor has been renamed *AppleScript Editor*.

The example scripts that once resided alongside Script Editor are not gone either. They're still located in */Library/Scripts* (an alias pointing to this folder was previously included in the */Applications/AppleScript* folder). Now, you can quickly navigate to them from within AppleScript Editor, by selecting *Open Example Scripts Folder* from the *Help* menu.

AppleScript Editor has received a number of other enhancements, too, in addition to its new name. First, you may notice some changes in formatting when you compile your scripts. Command names, parameter names, classes, and more, which were previously grouped under a generic *Application keywords* category of formatting, now have their own formatting attributes. These attributes can be adjusted in AppleScript Editor's *Preferences* window, under *Formatting*.

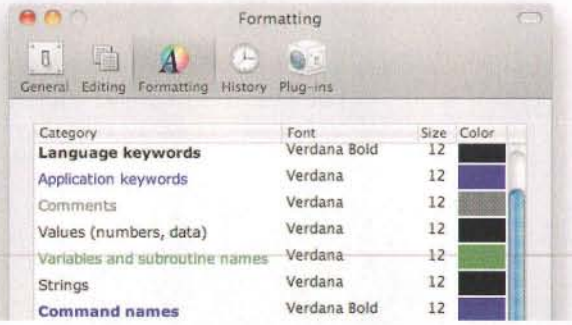


Figure 1. Formatting Options in AppleScript Editor's Preferences Window

Another visual change is that the event log and result panes at the bottom of the script window have been merged into a single pane, labeled *Event Log*. Within this new, merged pane, you now have the option to view a list of events, events and replies (replies are more clearly indicated than in previous versions), or the result. See figure 2.

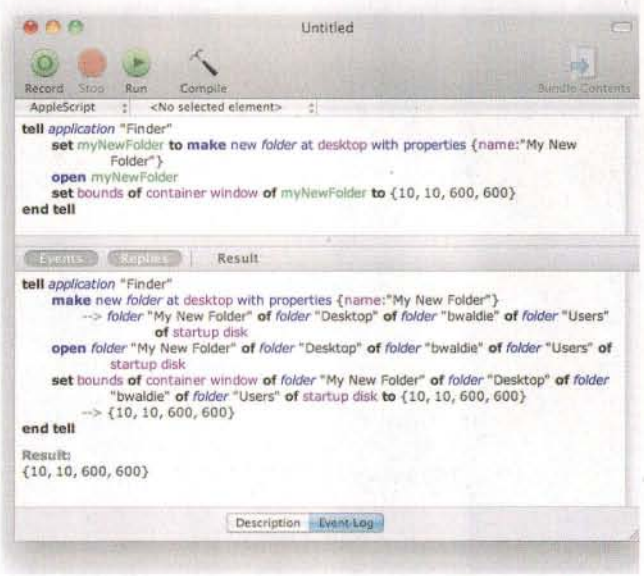


Figure 2. AppleScript Editor's Event Log, Displaying Both Events and Replies

The new event log pane also includes a number of other enhancements, which are sure to make it easier to both develop and troubleshoot your scripts. For advanced scripters, positioning the mouse cursor over an event in the log will now display a tool tip containing the raw Apple Event details for the event. Another *very* useful enhancement is that the event log now displays error messages, complete with both the error's description and number. This doesn't just occur when a showstopper error occurs either. It happens whenever any error is encountered, even if it's within a try statement. See figure 3 for an example of this.


ETYMOTIC
hf2
headset + earphones
Experience the unmatched fidelity
of this multiple award winning headset
• Highest noise-isolation of all in-ear earphones
• In-line mic for phone calls
• Remote control for music
Found in Apple stores and many other locations worldwide.
Online at Etymotic.com, Apple.com and Amazon.com.
Macworld Editors' Choice
iLounge Best of Show 2008
iPhone Matters 2008 iPhoneMatters.com Best of Show Macworld Expo
MacLife Editors' Choice
BEST OF 2008
Macworld awards 2009 WINNER
ETYMOTIC
The science of pure music
For more information on how we measure fidelity or to see our full line of high fidelity earphones and headsets visit www.bestipodsound.com
©2009. hf2 is a trademark of Etymotic Research, Inc. Earphones are covered by one or more of the following U.S. patents: #4,677,679, #4,763,753, #5,887,070, RE38351, RE40696 and other patents pending.

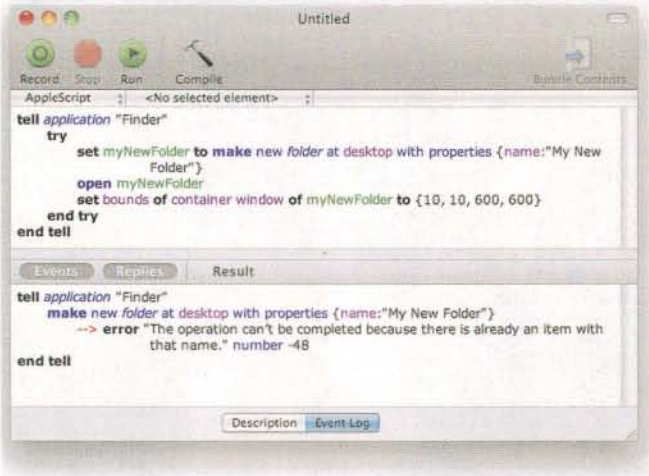


Figure 3. Error Logging in AppleScript Editor's Event Log

Another useful enhancement for experienced scripters is the new "tell" application pop-up menu, which can be enabled in the *Preferences* window, under *Editing*. See figure 4.

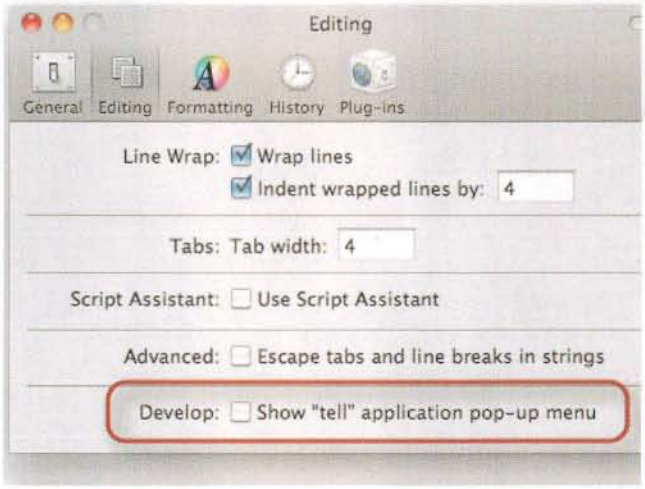


Figure 4. Enabling AppleScript Editor's "tell" Menu

Once enabled, a new pop-up menu appears in the navigation bar above the script window's editing area (if this bar isn't visible, select *Show Navigation Bar* from the *View* menu). This pop-up menu may be used to set the target of the script to a specified application. For example, setting this pop-up menu to the Finder will instruct AppleScript Editor that all code within the script should be sent to the Finder application, thus eliminating the need to implicitly use a "tell" statement. This capability would be useful for running quick application terminology tests during development, or for testing scripts that will be run from within an application (which serves as the script's target). For example, a FileMaker developer might use this feature

to write a script that will eventually run within a FileMaker database's *Perform AppleScript* script step. See figure 5.

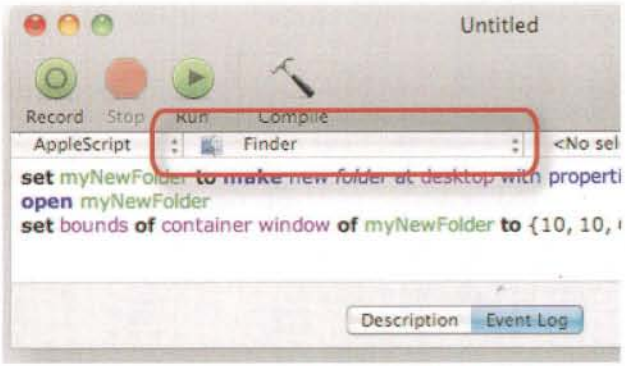


Figure 5. Targeting the Finder with AppleScript Editor's "tell" Menu

Changing a script's target with the "tell" application pop-up menu will only affect the script's behavior when it is compiled and run within AppleScript Editor, or when run from within the target application. If you change a script's target and then save the script as a stand-alone application, you will receive an error when you attempt to run it, as the script application becomes the target of the script.

Another more advanced enhancement is the AppleScript Editor's ability to run scripts on background threads. This means that you can now run multiple opened scripts at once. For example, you could be testing a lengthy script, while editing and testing a smaller, second script at the same time. You can also force a script to run on the main thread by holding down the *Control* key and selecting *Run in Foreground* from the *Script* menu. This may be necessary if your script interacts with a scripting addition that's not thread-safe.

Finally, another change to AppleScript Editor is that there is no longer an option to save a "flat" script application. Rather, all applications are saved as bundles, and are, therefore, not compatible with systems prior to 10.3.

AppleScript Utility

In previous versions of Mac OS X, an AppleScript Utility application, located in the */Applications/AppleScript* folder, provided a central location for managing various OS-level AppleScript settings. This application has been moved into the */System/Library/CoreServices* folder, and now operates as a background application, in order to provide backward compatibility with older scripts. Settings previously found in the AppleScript Utility, such as enabling, disabling, and configuring the system-wide script menu, can now be found in AppleScript Editor's *Preferences* window, under *General*. See figure 6.

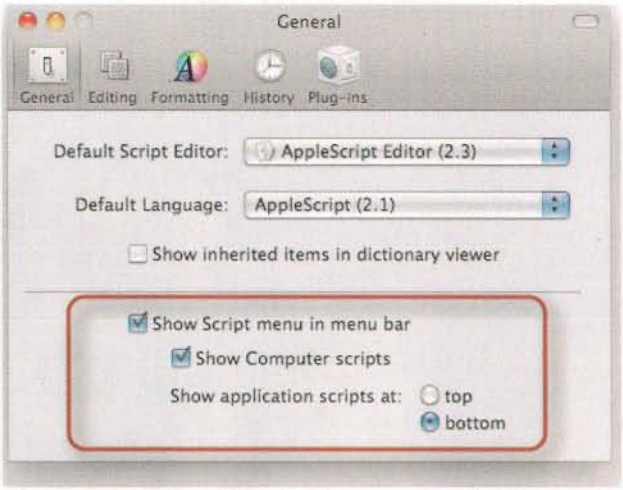


Figure 6. Script Menu and Other System-Wide Options in AppleScript Editor's Preferences Window

Folder Actions Setup

Another application that has been moved from */Applications/AppleScript* to the */System/Library/CoreServices* folder is Folder Actions Setup. This application, which allows you to enable and configure Folder Action scripts in Mac OS X, can now be launched by selecting *Folder Actions Setup...* from the Finder's contextual menu (see figure 7), or from the *Finder > Services* menu in the menu bar.

Figure 7. Launching Folder Actions Setup from the Finder's Contextual Menu

Folder Actions themselves have also received some upgrades. For one, they now support attaching Automator workflows directly to folders. Previously, a secondary AppleScript component was required to launch Automator workflows attached to folders. Another change, which I'm sure will prove a welcome enhancement for many, is that Folder Actions will now attempt to wait for an item to finish writing to the attached folder, before processing it. This is done by checking the size of the detected item until it remains static for at least three seconds. In the past, Folder Actions were notorious for triggering when an item first appeared in the folder, thus causing errors if the item was still being copied or saved.

Changes Under the Hood

Scripting Addition Calls

Snow Leopard introduces a number of other noticeable AppleScript changes. First, calls to scripting additions are now a bit finicky, and most will produce a *privilege error* when issued within an application "tell" statement. According to Apple, this is done for security purposes. Instead, scripting addition calls should be sent to the current application, i.e. the script itself or the process running the script.

To provide backward compatibility with older scripts, AppleScript automatically tries to capture this error, and redirect the problematic call to the current application. This is evident in figure 8's event log, where the *do shell script* call is first sent to the Finder, and then to the current application.

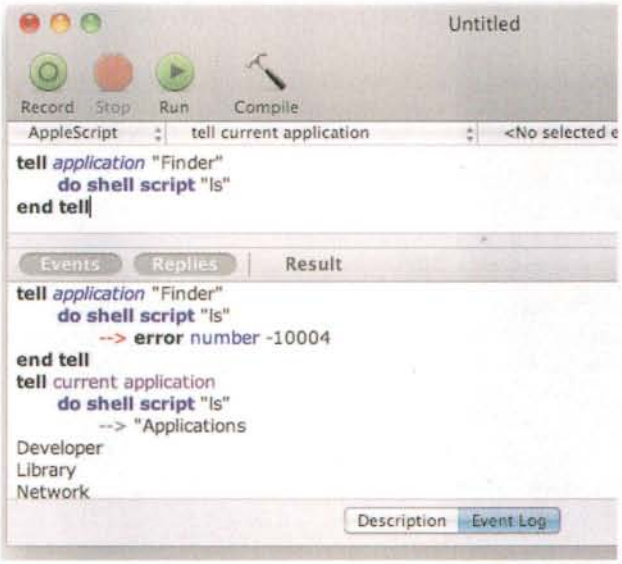


Figure 8. Redirected Scripting Addition Calls

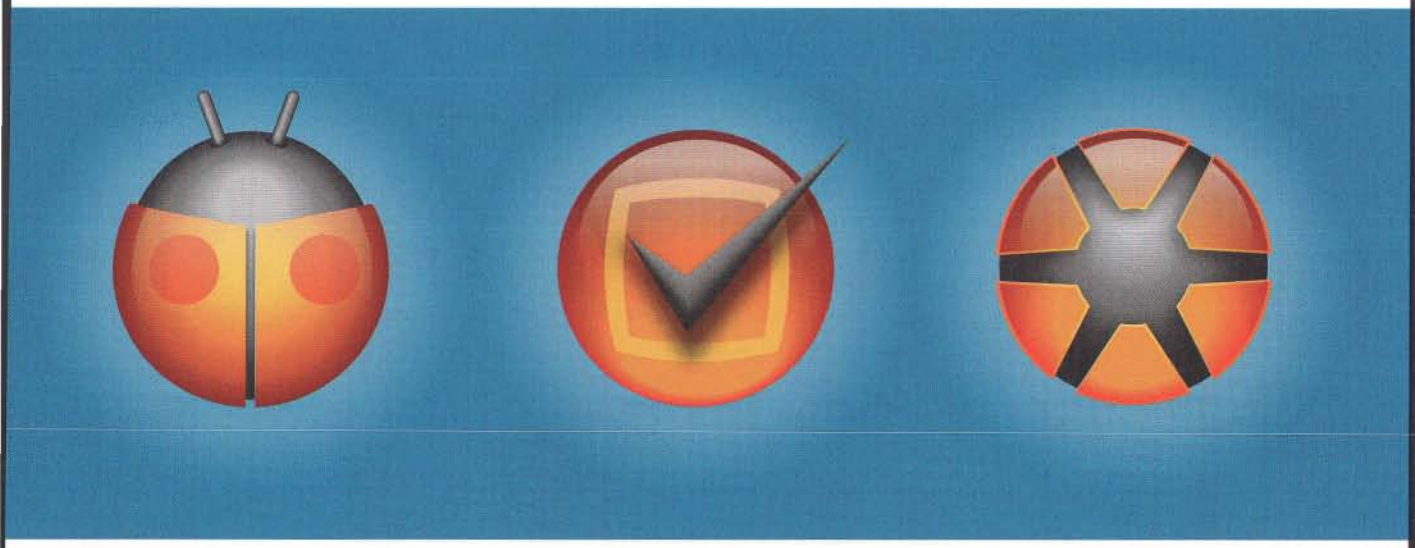
For optimal results and to ensure future compatibility, however, developers should try to avoid including calls to scripting additions within application "tell" statements. For example, change:

```
tell application "Some App"
    do some scripting addition command
    do some application command
end tell
```

... to:

```
some scripting addition command
```

Download free trials of TestTrack Pro, TestTrack TCM, and Surround SCM at:
www.seapine.com/mactech



Seapine ALM solutions for serious Mac OS X development

Stay on track with Seapine's Mac OS X-native development tools. Designed for the most demanding software development environments, Seapine's Mac OS X-native application lifecycle management (ALM) solutions are scalable, feature rich, team-based tools that can be used separately for superior issue tracking, test case management, and software configuration management–or seamlessly integrated for more efficient control of your software development process.

TestTrack Pro
Issue & Defect Management

- Track defects, change requests, feature requests, and other project-related issues.
- Tailor workflows, including events, states, and transitions, to your development process.
- Stay informed and on track with flexible reports, email notifications, and escalation rules.
- Create and link defects with failed test runs in TestTrack TCM for better traceability.
- Link defects and change requests to source code changes in Surround SCM and other SCM tools.

TestTrack TCM
Test Case Planning & Tracking

- Manage thousands of test cases, select sets of tests to run against builds, and process the pass/fail results using your development workflow.
- Ensure all steps are executed, and in the same order, for more consistent testing.
- Know instantly which test cases have been executed, what your coverage is and how much testing remains.
- Track test case execution times to more accurately estimate the time required to test applications.

Surround SCM
Configuration Management

- Control who changes source files, and track what has changed and when.
- Leverage file-level workflow to track and manage the state of individual files.
- Facilitate parallel development with advanced virtual branching, private branches, and workflows.
- Notify team members of new files, assignments, and changes by email.
- Quickly access the latest files with shadow folders, hyperlinks, and Finder integration.

Seapine Software™

QA Wizard® Pro Automated Testing

Seapine CM® Change Management

Surround SCM® Configuration Management

TestTrack® Studio Test Planning & Tracking

TestTrack® TCM Test Case Management

TestTrack® Pro Issue Management

```
tell application "Some App"
    do some application command
end tell
```

Alternately, you can implicitly target the current application in your scripting addition call. For example:

```
tell application "Some App"
    tell current application to do some scripting addition
command
    do some application command
end tell
```

Date Handling

Coercions from strings to dates are also a bit finicky in Snow Leopard. Previous versions of AppleScript were fairly forgiving when attempting to coerce a string to a date. For example, in Mac OS X 10.5, the following script ran successfully on my machine:

```
set theDate to "03, September, 2009"
date theDate
-> date "Thursday, September 3, 2009 12:00:00 AM"
```

In Snow Leopard, however, the same script fails with the following error message:

```
-> error "Invalid date and time date 03, September, 2009 of
«script»." number -30720
```

To prevent these types of errors, Apple recommends ensuring that any string being coerced to a date match one of the system date formats found in System Preferences, under *Language and Text* (formerly *International*) > *Formats*. For example:

```
set theDate to "September 3, 2009"
date theDate
-> date "Thursday, September 3, 2009 12:00:00 AM"
```

Standard Additions Changes

Only a few minor changes have been made to the Standard Additions scripting addition. The *do shell script* command now provides better error reporting, and the *say* command includes some interesting new parameters for controlling the pitch, modulation, and rate of spoken text. So, you can now have your script sound more like Darth Vader, for example:

```
say "No, 'Luke'! 'I' 'am' your father!" using "Bruce" speaking
rate 200 pitch 25 modulation 20
```

System Events Changes

The System Events background application has also received a few updates, such as providing access to screen savers. And, menu bar transparency is now accessible via scripting. For example:

```
tell application "System Events"
    set translucent menu bar of current desktop to false
end tell
```

Application Scripting Changes

As with any major system update, scripters should also be on the lookout for AppleScript terminology changes in updated Snow Leopard applications. As one example, QuickTime Player has received some fairly hefty enhancements in Snow Leopard. As a result, its AppleScript terminology has changed substantially too. Users with existing QuickTime scripts should take a careful look at this updated terminology, and make any necessary adjustments to their scripts.

AppleScript Studio Changes

Perhaps the most significant AppleScript change in Snow Leopard involves AppleScript Studio, which has officially been deprecated. Yep, you read that correct, deprecated, as in obsolete and discontinued.

AppleScript Studio was a subset of features in Apple's Xcode development environment, which allowed scripters to build AppleScript-based applications, complete with Cocoa interfaces (created in Interface Builder).

For those scripters who have created countless AppleScript Studio applications, don't freak out just yet. First, Apple says that AppleScript Studio is still supported in Snow Leopard, and existing applications should continue to function. You should also be able to continue opening your existing AppleScript Studio projects and editing them in Xcode.

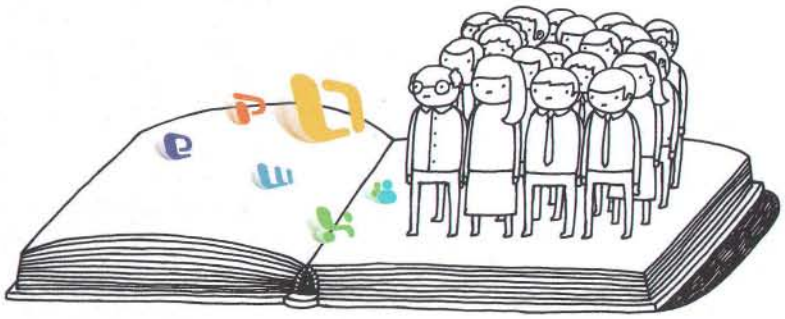
You will not, however, be able to create new AppleScript Studio projects in Xcode, as the templates for doing so have been removed. In addition, the handy AppleScript palette you're used to seeing in Interface Builder is now hidden by default. You can re-enable it, however, by running the following command in Terminal:

```
defaults write com.apple.InterfaceBuilder3
IBEnableAppleScriptStudioSupport -bool YES
```

Another option may be to install a second, older version of Xcode, alongside Snow Leopard's Xcode, thus allowing you to continue developing new AppleScript Studio applications. Installing multiple versions of Xcode was supported in the past, though I have yet to test it for myself in Snow Leopard, and I can't say whether it will continue to be supported in the future.

Regardless, there will come a day when AppleScript Studio is no longer supported. So, developers should start thinking about this now, and begin creating new applications and migrating existing ones to AppleScript Studio's successor – AppleScriptObjC – as soon as possible. AppleScriptObjC is a new framework in Mac OS X, which provides a bridge between AppleScript and the Objective-C runtime, thus allowing AppleScript to interact with any Cocoa framework.

For scripters without a background in Cocoa and Objective-C, AppleScriptObjC will take some time to master. However, there are some resources that can help get you started. For one, Xcode includes a new project template, named *Cocoa-AppleScript Application*. See figure 9. There's a *Cocoa-AppleScript* Automator action template too, for those Automator developers.



Everyone on the same page

Work together. Different machines? Different platforms? No matter. You can all speak the same language.

SimplifyYourWork2008.com

Microsoft® Office:mac 2008

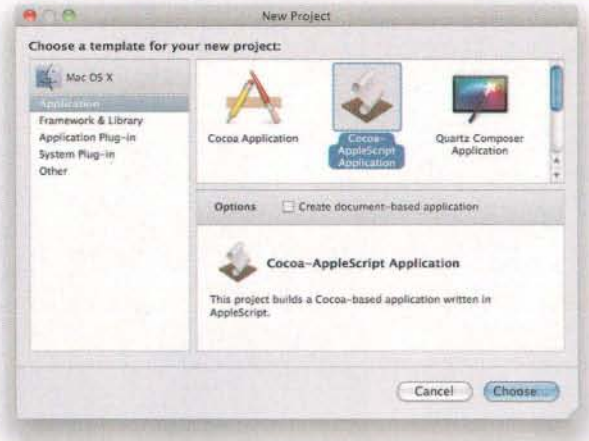


Figure 9. Creating a Cocoa-AppleScript Application in Xcode

You can also browse the Mac OS X reference library for Snow Leopard, which, at the time I wrote this column, included some preliminary AppleScriptObjC release notes, as well as a brief, though informative, *Objective-C/AppleScript Quick Translation Guide*. If you're new to Objective-C and Cocoa, I'd also recommend checking out the *Cocoa Fundamentals Guide* and the *Introduction to The Objective-C 2.0 Programming Language*. All of these documents are available on the Apple Developer Connection website at http://developer.apple.com.

In Closing

Overall, the AppleScript changes in Snow Leopard come as welcome improvements. That said, some changes are likely to cause compatibility issues with existing scripts. As with any Mac OS X update, AppleScript developers should take the time to test existing scripts thoroughly, and make any necessary changes to ensure compatibility.

AppleScriptObjC will likely require a bit more time and effort to upgrade existing scripts, and there *will* be a learning curve. That said, scripters adopted AppleScript Studio fairly quickly when it was first introduced, and I have no doubt that they will do the same for AppleScriptObjC. Once the migration is complete, however, the power and reach of AppleScript will be greater than ever before.

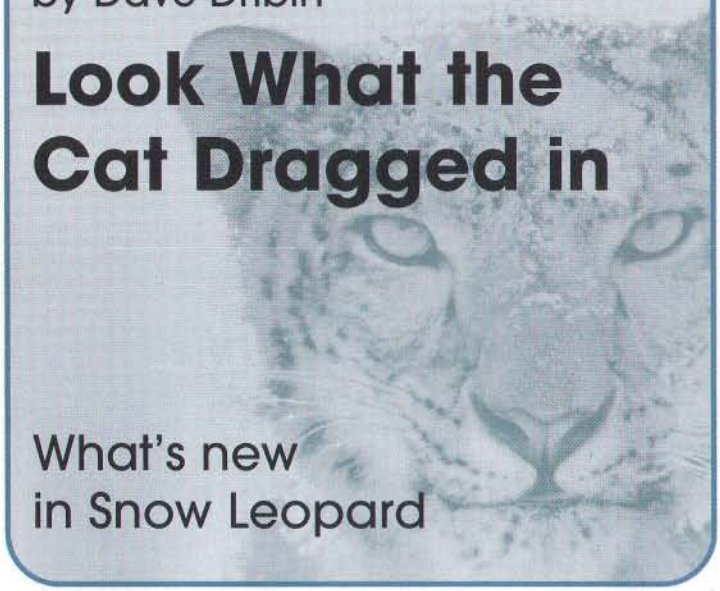
Until next time, keep scripting!

About The Author

Ben Waldie (ben@automatedworkflows.com) is president of Automated Workflows, LLC (www.automatedworkflows.com), a company offering AppleScript, Automator, and workflow consulting services to Mac-based businesses. For years, Ben has developed professional automated solutions for companies such as Abercrombie & Fitch, Adobe Systems, Apple Inc., CNN, Microsoft, NASA, PC World, and Time Magazine. Ben is the author of "Automator for Mac OS X 10.5 Leopard Visual QuickStart Guide" (Peachpit Press) and "AppleScripting the Finder", has written AppleScript and Automator content for Apple.com, Macworld, MacTech, MacScripter.net, and X-Ray Magazine, and is the host of the "Mac Automation Made Simple" video podcast (Peachpit Press). Ben has also released hundreds of Automator actions for use with Adobe Illustrator, InDesign, Photoshop, FileMaker, QuarkXPress, Twitter, and more.

by Dave Dribin

Look What the Cat Dragged in



What's new in Snow Leopard

Another Release Already?

It's hard to believe that it's been nearly two years since Mac OS X 10.5, Leopard, was released. The next major version of Mac OS X, version 10.6, code named Snow Leopard, is now upon us. While Apple may have originally touted that Snow Leopard contained only one new feature, this release is as packed full of developer goodness.

Blocks

It's probably best to work our way up from the bottom and discuss what's new at the language level, first. In Leopard, we got Objective-C 2.0 and garbage collection. With Snow Leopard, we get another very big feature: *blocks*. A block, sometimes called a *closure* or *lambda* in other languages, is similar to a function pointer, except that is can be defined inline and is captures the local stack variables for later use. Blocks can be used in plain C as well as Objective-C, and their syntax is very similar to function pointers. Here's a simple example of function pointers in standard C:

```
void printJoe(void)
{
    printf("My name is Joe.\n");
}

void printJane(void)
{
    printf("My name is Jane.\n");
}

void printNames(void)
{
    void (*printName)(void) = NULL;

    printName = printJoe;
    printName();
    printName = printJane;
    printName();
}
```

The `printName` variable is a function pointer. The strange syntax is needed to define its expected return type and arguments. In this case, `printName` is a function pointer that returns void and takes no arguments, a void parameter. It's a pointer just like any other pointer, so we can set it to `NULL`, just like any pointer. You can assign it any function that matches its signature, i.e. any function that returns nothing and takes no parameters. In the example above, we first assign the `printJoe` function to the `printName` variable. Notice that we do not use parentheses on `printJoe`, since that would actually call the function. Once we assign a non-`NULL` value to the `printName` variable, we can use it to call the function it's pointing to, and the syntax looks like an ordinary function call. Thus, this statement calls the function being pointed to:

```
printName();
```

In the first case, this calls the `printJoe` function. We then assign it again, this time to `printJane`, and call it again. If we called the `printNames` function, we would get the following output to the console:

```
My name is Joe.
My name is Jane.
```

Function pointers in C are a rather obscure feature that doesn't get used very often. There just aren't a lot of real world uses for them.

Blocks are similar to function pointers, but a *lot* more flexible. The syntax for block variables is similar to function pointer variables, except that you use a caret (^) instead of a star (*). Here's how we could rewrite `printNames` using blocks:

```
void printNames(void)
{
    void (^printName)(void) = NULL;

    printName = ^{
        printf("My name is Joe.\n");
    };
    printName();

    printName = ^{
        printf("My name is Jane.\n");
    };
    printName();
}
```

The `printName` variable is declared using similar syntax. Creating and assigning a block to the `printName` variable again uses ^.

This demonstrates that you can create a block of code inline, rather than factoring it out into a separate function. Blocks can also take arguments and have a return value:

```
void doMath(void)
{
    int (^multiply)(int x, int y);
    multiply = ^(int x, int y) {
        return x*y;
    };

    printf("Multiply: %d\n", multiply(3, 5));
}
```

The real benefit of blocks, however, is their ability to capture local stack variables and use them:

```
void doMath(void)
{
    int (^multiply)(int x);
    int y = 5;
    multiply = ^(int x) {
        return x*y;
    };

    printf("Multiply: %d\n", multiply(3));
}
```

In this case, the y variable is captured and can be used inside the block. This is a contrived example, but we will see an example where the benefit of this becomes more readily apparent.

The previous examples show how blocks can be used from straight C, but many Objective-C APIs have been updated to take advantage of blocks. For example, `NSArray` has a new method to enumerate its items using a block:

```
NSArray * colors = [NSArray arrayWithObjects:
                    @"Red", @"Green", @"Blue", nil];
[colors enumerateObjectsUsingBlock:
 ^(id color, NSUInteger idx, BOOL *stop) {
    NSLog(@"Color: %@", color);
}];
```

Here, we specify a block that gets invoked for every element in the array. This is actually a bit more verbose than the `for…in` syntax added in Leopard, but there certain things you can do better by enumerating with blocks. For example, you can enumerate in reverse order efficiently. There's also a way to enumerate items concurrently, allowing you to fully exploit multiple cores, if there are no dependencies between each iteration.

Blocks are really good for asynchronous actions. A common pattern in Objective C is to provide a selector to be called when an asynchronous action is finished. This is used in sheets where the selector is called when the sheet is finished. Blocks make this kind of idiom much easier, and `NSSavePanel` and `NSOpenPanel` now have a block-based API when using sheets. Here's example code that runs an open panel for text files:

```
- (IBAction)openTextFiles:(id)sender
{
    NSOpenPanel * panel = [NSOpenPanel openPanel];

    NSArray * fileTypes = [NSArray arrayWithObject:@"txt"];
    [panel setAllowedFileTypes:fileTypes];

    [panel setAllowsMultipleSelection:YES];
    [panel beginSheetModalForWindow:self.window
                  completionHandler:^(NSInteger result) {
        if (result == NSOKButton) {
            NSArray * filenames = [panel filenames];
            for (NSString * filename in filenames) {
                [self doSomethingWithFile:filename];
            }
        }
    }];
}
```

The beginSheetForModalWindow: method takes a block as the final argument. The sheet is started and this method returns right away. When the user dismisses the sheet, the completion handler is called. This block takes a single integer argument corresponding to which button was press. In this code, we take some action if the user pressed the Open button. Remember what I said about blocks capturing local variables? This is shown by the ability to use the `panel` stack and `self` variables inside the block.

Concurrency

Grand Central Dispatch

Grand Central Dispatch, or GCD, is a new framework to help developers fully utilize the multiple CPU cores shipping on all Macs. It is built upon blocks and adds what's known as *dispatch queues*. Dispatch queues are a queue of blocks that take no parameters and return nothing. As you enqueue blocks to a queue, they get executed. Here is a simple example showing you how to add a block to a queue:

```
dispatch_async(queue, ^{
    NSLog(@"Async");
});
```

This method is asynchronous, meaning it returns right away. The block is then executed on the queue, which may even be a different thread.

GCD provides global queues for dispatching blocks to background threads. Blocks added to the global queue also run concurrently with each other. Here's how you would get the global queue, with the default priority:

```
dispatch_queue_t queue =

dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT,
0);
```

Combining these two lines is how you get very easy concurrency.

GCD also provides us with a queue that corresponds to the main thread:

```
dispatch_queue_t queue = dispatch_get_main_queue();
```

Blocks added to the main queue get executed on the main thread. Blocks are also executed serially on the main queue, meaning there is no concurrency and they execute strictly in the order they are added. Using the main queue can replace the old `performSelectorOnMainThread:` method to shuttle work over to the main thread. It is important to only access AppKit from the main thread, so this makes dealing with these circumstances even easier.

So how can these be used to help bring concurrency to your application? Say you have a method that can take some time to execute. If you executed this on the main thread in response to a button press, there's a good chance you will block the UI thread. When this happens, the user sees the spinning pinwheel icon, and the application becomes unresponsive. Here's some sample code:

```
- (IBAction)buttonPressed:(id)sender
{
    NSURL * url = [NSURL URLWithString:[_urlField stringValue]];
    NSString * string = [self getStringFromUrl:url];
    [_textField setStringValue:string];
}
```

In this code, if the `getStringFromUrl:` method takes a long time to execute because it performs a network operation, it could block the UI thread. Since this is undesirable, the way to deal with this is to execute `getStringFromUrl:` on a background thread. Of course, we need to set the result on the main thread, since we are putting the result in a text field. Using GCD and blocks, this becomes very easy to do:

```
- (IBAction)buttonPressed:(id)sender
{
    dispatch_queue_t concurrentQueue =
dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0);
    dispatch_queue_t mainQueue = dispatch_get_main_queue();

    NSURL * url = [NSURL URLWithString:[_urlField
stringValue]];
    dispatch_async(concurrentQueue, ^{
        NSString * string = [self getStringFromUrl:url];
        dispatch_async(mainQueue, ^{
            [_textField setStringValue:string];
        });
    });
}
```

First, we grab the URL from the text field on the main thread. Then, we use a global concurrent queue to call the `getStringFromUrl:` method on a background thread. Finally, we use the main queue to update the UI. This again shows the power of blocks being able to access variables on the stack. For example the `url` variable is available in the global queue block and the `string` variable is available on the main queue block.

Writing the equivalent code without GCD and blocks is certainly possible, but it ends up being a lot more verbose. You would need to factor out the code into separate methods and create a thread yourself. Also, you would need to pass the objects as arguments to the methods. In more complicated cases, you often need to create instance variables because the `performSelector` methods only allow you to pass one or two objects. Blocks and GCD eliminate all of this hassle.

There is quite a bit more to GCD. You can create your own queues, group blocks together, and even replace locks with dispatch queues; however, this should at least provide a hint of how much easier it is to make your application concurrent with GCD.

Mac Universal

Freehand Capture

Mail images

Full Screen capture

Multiple capture

Crop Freehand area

Torn Edge

DOM capture

iSight Capture

Perspective

Polygonal Capture

Blur Tool

Capture

Text tool

Timed capture

Custom Collections

Print image

Vector Annotations

Circular capture

Upload to FTP

20 Different Sprays

Tag images

Pencil tool

Effects

Quicklook Support

Paint tool

Share

Annotate

Built-in Trash

Skew

14 Core image Filters

Smart Collections

Straight arrows

Curved Arrows

Stamps

Image as annotation

Image Description

Image Database

Menu capture

Edit before capture

Sawtooth Edge

Organize

Object capture

Flowchart Shapes

Customize capture shortcuts

Glossy Callouts

Upload to Flickr

Multi Screen Capture

Full Webpage Capture

NOMINATION Macworld awards 2009 NOMINATION

Macworld Editors' Choice ★★★★

Macworld ★★★★1/2

"A great replacement and enhancement for your Mac's screen-capture features, Voila is ideal for tutorials, business presentations or even basic photo editing - Macworld

Still want more reasons to try Voila - The best Mac Screen Capture Software?
Visit: www.globaldelight.com/voila and download the free 30 day trial

NSOperation

`NSOperation` and `NSOperationQueue` were added in Leopard to help with writing concurrent code, and in Snow Leopard they are written on top of GCD. This means that `NSOperationQueue` now performs better than it did in Leopard. There is a new block operation subclass, so you can easily turn a block into an operation:

```
NSOperation * operation =
    [NSBlockOperation blockOperationWithBlock:^{
        // Do long running task
    }];
```

There is also a new main operation queue, which is similar to the GCD main dispatch queue. The main operation queue runs its operations serially on the main thread:

```
NSOperationQueue * queue = [NSOperationQueue mainQueue];
[queue addOperationWithBlock:^{
    // Run on the main thread
}];
```

This also demonstrates the new `addOperationWithBlock:` method, which eliminates the need to deal with an `NSOperation` subclass at all, for simple cases.

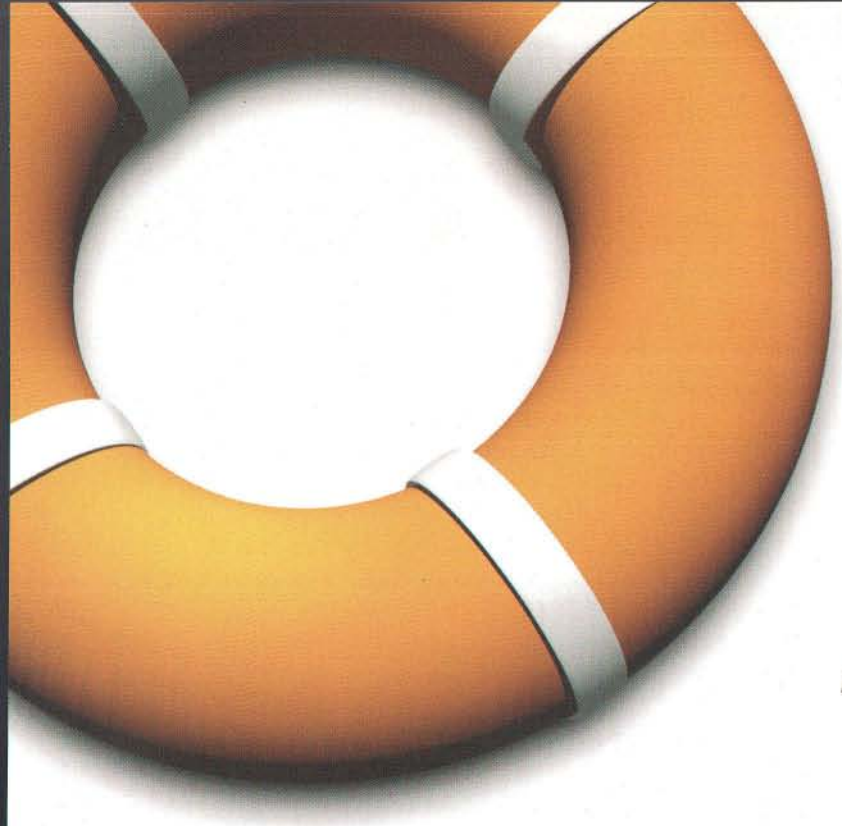
In Snow Leopard, the behavior of concurrent operations has changed. A concurrent operation is one that overrides the `-isConcurrent` method to return `YES`. Concurrent operations act a little differently in Snow Leopard. The `-start` method is always called on a background thread. If you used concurrent operations to start asynchronous actions on the main thread, such as starting an `NSURLConnection`, this may trip you up.

64-Bit

In Leopard, AppKit was available in 64-bit mode for applications that needed the extra memory, however 32-bit applications were still the default. In Snow Leopard, 64-bit mode is preferred to 32-bit on hardware that supports it, and nearly all system applications are available in 64-bit mode, including the Finder, which was rewritten in Cocoa.

There are additional benefits to 64-bit mode over 32-bit beyond access to more memory. Because 64-bit mode is new, Apple was able to break backwards compatibility for the Objective-C application binary interface (ABI) to provide better performance. One example of this is the fact that arguments to functions and methods are passed in registers wherever possible. In 32-bit mode, arguments were always passed on the stack. Using registers can give every function and method call a performance boost. These benefits are nothing new to Snow Leopard, but as all system applications are now running in 64-bit mode, they may feel a bit snappier for this reason alone.

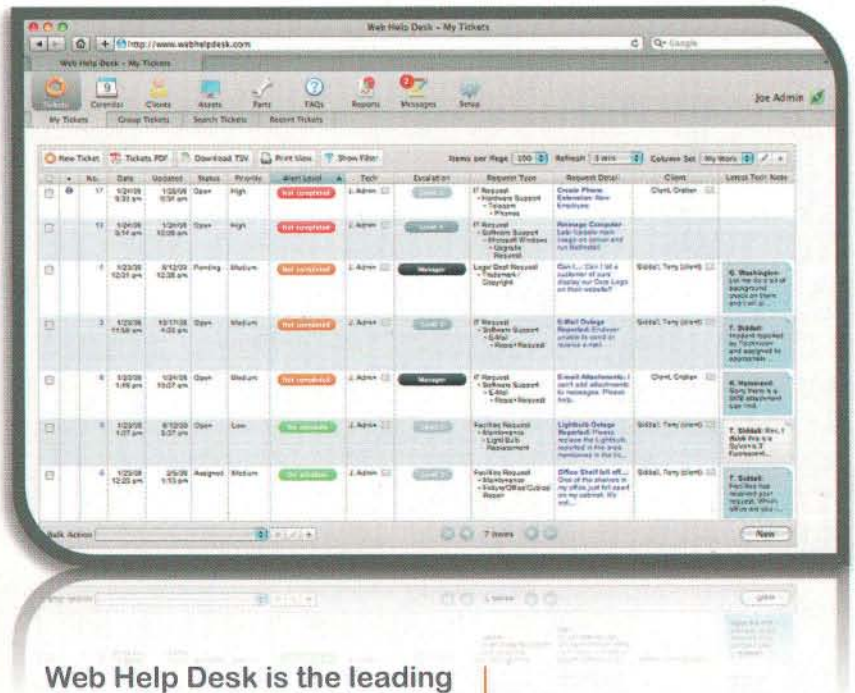
One important point to note is that if you are writing an application that runs on 10.5 and 10.6, you will probably still want to write in 32-bit mode on 10.5. The reason for this is that it's best to use whatever the system default is, unless you have a good reason not to. If your application is the only 64-bit



Help Has Arrived.

Web Help Desk

Powerful Software for Service Management



Web Help Desk is the leading cross platform service management solution for support professionals seeking to simplify and control their increasingly complex service management environments.

Contact us today to learn more.
www.WebHelpDesk.com

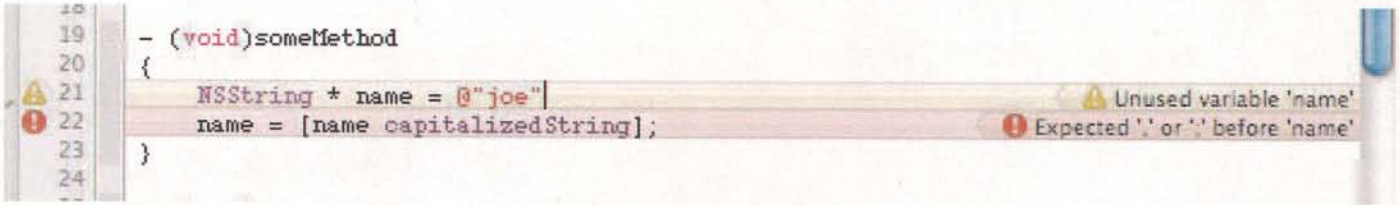


Figure 1: New error bubbles

application on a 10.5 machine, your application will be using a lot more memory because it has to use 64-bit versions of all the system frameworks. Much of the memory for the system frameworks is shared amongst all running applications, and if your application is the only 64-bit application, then it cannot take advantage of the memory sharing.

Ideally, your application should run as 32-bit on 10.5 and 64-bit on 10.6. There's actually some trickery you can do in your Info.plist so that your application works this way. Here is a snippet of XML to show how you could set this up:

```
<key>LSArchitecturePriority</key>
<array>
    <string>x86_64</string>
    <string>i386</string>
    <string>ppc</string>
</array>
<key>LSMinimumSystemVersionByArchitecture</key>
<dict>
    <key>x86_64</key>
    <string>10.6</string>
</dict>
```

By using the `LSMinimumSystemVersionByArchitecture` key, you're telling the OS to use 64-bit mode (`x86_64`) only on 10.6 and later. Thus 10.5 will still use 32-bit mode (`i386`).

Of course, this means that your applications will need to actually support 64-bit mode. You don't want to be the only 32-bit application running on Snow Leopard. So if you haven't already, build and test your applications for the `x86_64` architecture.

Core Data

Some of the Core Data improvements include:

lightweight migrations,

Spotlight integration, and

new fetch operations to tune performance.

Lightweight migrations mean that simple changes to your schema can be handled without any code at all and no mapping model. This is a welcome addition since changing

schemas are a part of life. You can still drop back to Leopard's migration for complicated schema changes, but this takes the hassle out of the easy changes.

Core Data now has built-in support for Spotlight integration. Spotlight requires that every "record" have its own file on the file system. This has traditionally been problematic for Core Data databases, since all record data is stored in a single file. To rectify this situation, Core Data provides a mechanism to generate the Spotlight metadata files and keep them automatically up-to-date with the Core Data database.

There are a whole host of new fetch options to help you tune your fetches. The first is the ability to fetch in batches. Previously, you had only two options: fetch the results as managed objects or fetch as faults. For large data sets, where only a few objects are displayed in the UI, fetching all results as managed objects uses more memory than needed. On the other hand, fetching results as faults means every item will need to be faulted in as it is displayed. Using batch fetching allows you to fetch a subset of the results as managed objects. This is the best of both worlds. By tuning the batch size, you can ensure that just enough data to be displayed in the UI is fetched, but still be memory efficient.

Another technique for increasing fetch performance is to use partial faults. If the UI is only displaying a few of the attributes of a managed object, it is wasteful to populate all attributes from the database. Partial faults allow you to read in only the data you need, thus saving I/O and memory.

One final interesting fetch option is the ability to fetch results as a dictionary. This can be very handy for read-only interfaces. Not only does this provide the ability to only pull back the data you are interested in, similar to partial fetches, but it also allows you to use aggregates and distinct values to push more processing down to the SQLite level.

AppKit Updates

The pasteboard and system services have gotten a nice upgrade in Snow Leopard. You now interact with the pasteboard using standard Universal Type Identifiers (UTIs). This provides consistency with the rest of the operating system, where UTIs have been used throughout, instead of the old constants like `NSStringPboardType`. You can also place multiple items on the pasteboard, which is something only Carbon applications could do. While the old API is still available, there are new methods on `NSPasteboard` to support these new features, and any new application should use them.

System services have also gotten an overhaul for Snow Leopard. System services encompass the Services menu in each application. In previous versions of Mac OS X, the Services menu showed *all* services, even ones that were disabled. Thus, the menu got very large and unwieldy. Also, a service could register a hot key that would, in effect, become a global hot key, because services are available in

all applications. Unfortunately, the user had no way to disable or change these shortcuts. Both deficiencies have been addressed in Snow Leopard.

The Services menu now only lists services that you can actually use. Those that are not valid for the current context are not shown at all. Furthermore, the user can edit the keyboard shortcuts in System Preferences. As an extra bonus, services can show up in the context menu. This provides a method to extend the functionality of applications without resorting to a plug-in. This will be especially useful for the new Finder, which won't load any plug-ins for 10.6.

Finally, some views that have seen improvement in Snow Leopard are `NSImage`, `NSBrowser`, and `NSCollectionView`. Be sure to read the release notes if you want to find out more information.

Developer Tools

What OS upgrade would be complete without updates to the developer tools? The biggest winners in Snow Leopard are Xcode and Instruments.

Xcode gets a bit of a face-lift. The error bubbles introduced in Leopard have been refined to be less intrusive, as shown in Figure 1. The build results window has been completely revamped to help you focus on the results and also deal with large build logs.

The code completion has been improved and is more context-aware. The Open Quickly dialog seems to be much faster.

You can rename a project. Since this isn't as easy as renaming the .xcodeproj file, Xcode now renaming provides you with a wizard so you can rename other parts of your project that may also need to be renamed such as targets and menu items, too.

You can also fill in the company name as a per-project setting. While Xcode 3.1 used your company name from your Address Book contact, this allows for more precise control. This is great for people who work for multiple organizations. So there's really no more excuse to have `__MyCompanyName__` in your header comments.

Xcode now has built-in support for the Clang Static Analyzer. The Clang Static Analyzer runs static analysis on your code to find many common errors that the compiler does not normally find, even with all warnings enabled. The analysis results show up in the Build Results window, just like other errors. Running the static analyzer for the first time is very eye opening. It'll find all sorts of bugs that have most likely been lurking for a while, and I highly recommend you get into the habit of running the static analyzer periodically.

Instruments has also been vastly improved in Snow Leopard. The following instruments have been added:

Time Profiler
Dispatch
Multicore
Threads
Zombies
Object Graph
Garbage Collection

The Time Profiler is a statistical sampler, similar to Shark, and allows you to pinpoint performance hotspots. It is very low impact and provides more precise timing data by sampling in the kernel.

The Dispatch, Multicore, and Threads instruments are new instruments that help get the most out of GCD and threads. If you're trying to make your application more concurrent and are not seeing the benefits you expect, these instruments can help you.

The Zombies instrument help you find places were you use objects after they are deallocated. Using over-released objects typically cause crashes, and can be hard to track down. The system helps you track over-released objects by turning your object into a *zombie object*, instead of actually freeing the memory. Any messages called on the zombie object trigger Instruments, complete with a stack trace. Coupled with the Object Alloc instrument, you can know which object was being over-released. This instrument takes place of the `NSZombiesEnabled` environment variable and makes it easier to track zombies with a nice GUI.

Finally, the Object Graph and Garbage Collection objects help you tune your garbage collection enabled applications. While garbage collection does help deal with memory management, it's still possible to leak memory and use more memory than you intend. Usually this is due to an object being referenced after it is no longer needed. The Object Graph instrument helps pinpoint these cases so you can see what is preventing an object from being collected and freed.

Conclusion

This is a relatively brief overview of what's new for developers in Snow Leopard, and it's just the tip of the iceberg. Be sure to read all the release notes to learn about all of the goodies.

About The Author

Dave Dribin has been writing professional software for over eleven years. After five years programming embedded C in the telecom industry and a brief stint riding the Internet bubble, he decided to venture out on his own. Since 2001, he has been providing independent consulting services, and in 2006, he founded Bit Maki, Inc. Find out more at <http://www.bitmaki.com/> and <http://www.dribin.org/dave/>.

Network, Server and Appliance Monitoring.

For Mac OS X

Xserve RAID

Xserve (Intel & G5)

Airport

Lithium Network Monitoring Platform

Lithium can now monitor your Xserve, Xserve RAID, Qlogic switches, Airports, Mac OS X Server... and everything else in your network.

www.lithiumcorp.com

lithium Corp.

Packaging for Enterprise Deployment

Building Installer packages for mass unattended distribution

By Greg Neagle, MacEnterprise.org

MacEnterprise.org
Mac OS X enterprise deployment project

Introduction

In past MacTech articles, we discussed reasons Mac OS X systems administrators might need to package or repackage software for deployment in their organization. Some common reasons are:

Packaging internally-developed software or tools

Capturing organization-specific changes or additions (licensing, configuration) into a package

Packaging third-party software that is distributed without an Installer package

Repackaging into a format compatible with your distribution mechanism

Repackaging software that won't install "silently" (i.e., prompts the user for info, or launches GUI applications)

In an ideal world, only the first two reasons would require a systems administrator to create an installation package (and if you can train your internal developers on packaging, maybe you can cross the first one off, too!). For the other scenarios, it would be more desirable to just take the installation package provided by your third-party software vendor and use it without having to repackage the software. This month, we'll talk about some of the reasons third-party installers don't work in enterprise environments and provide suggestions on how third-party software vendors could make their installers more enterprise-friendly.

Previous MacEnterprise columns have been targeted towards systems administrators in enterprise-scale organizations. This month's column should also be of interest to systems administrators, but I hope MacTech readers who are software developers find it relevant as well.

If you are a software developer, why should you bother making your installer friendly to enterprise environments? The number one reason is that if you do, you'll sell more of your product into an enterprise environment. Make it too hard to deploy in an enterprise, and systems administrators will look elsewhere and recommend your competitor's solutions.

Reason two: if your installer is not enterprise-friendly, systems administrators will almost certainly have to repackage your software in order to deploy it to their users. This introduces an opportunity for errors to creep into the install process. It's much easier to support your own software when you know it was installed with your own installer – if organizations have to repackage your software for installation, you've just introduced a new variable to support, or many, many new variables, as each organization that repackages your software may do it slightly differently.

Definitions

What does it mean when we refer to an installer as being enterprise-friendly? It's actually quite simple. An installer is enterprise-friendly when a system administrator can use standard mass-deployment tools to install your software on many machines automatically, and when, once installed this way, the software works as expected for all users of a given machine with no additional post-install configuration that cannot be done as a non-privileged user (i.e., a user without administrator rights).

Now you're wondering: "what does this author mean by 'standard mass-deployment tools'?" Again, a simple answer: on OS X, a standard mass-deployment tool is one that uses the Apple package format. This includes Apple Remote Desktop, LANrev, the Casper Suite, the KBOX, and many other commercial tools. All of these tools can use the Apple package format to install software remotely to a large number of managed machines.

Simple advice

The first, cheapest, simplest, and most important thing you can do as a software developer to ensure your software's installer is enterprise-friendly is to use an ssh session and the command-line installer tool (/usr/sbin/installer) to install your software on a remote machine. Test the install with no-one logged in as a GUI user, and for extra points, test the install with a GUI user logged in.

Some more details:

Copy the installer to the target machine in any way that is convenient – use scp to copy a disk image or tar or zip archive; or log on directly to the target machine and copy the installer from the web or a file server. Put it someplace readily accessible like /Users/Shared.

If a GUI user is logged into the target machine, log that user out.

Use ssh to login to the target machine as root or as an administrative user that has sudo rights.

```
> ssh gneagle@aquaman
Password:
Last login: Thu Feb  5 15:16:59 2009
(aquaman) gneagle [201] %
```

If you've logged in as an administrative user, use sudo to become root, and try to install your software:

```
(aquaman) gneagle [201] % sudo -s
Password:
[aquaman:/] root# cd /Users/Shared/
[aquaman:/Users/Shared] root#
[aquaman:~] root# installer -target / -pkg MyApp.pkg
```

If you support Tiger, test with a target machine running Tiger as well as Leopard. Watch what happens on the target machine while the install is happening. No windows or dialogs should appear; no applications should launch. You should see no visible indication that anything is happening.

Assuming the installation was successful and silent, now, using the GUI on the target machine, log in as a non-admin user and verify your software works as expected. Do the same as an administrative user.

Repeat the experiment, this time with a GUI user login on the target machine.

If your software can be installed silently with this method, there is no visual indication that anything is happening when the install is occurring, and your software functions as expected when installed this way: congratulations! There's a very good chance your installer is enterprise-friendly!

Common pitfalls

As it turns out, it's not hard at all to make an Installer package that is enterprise friendly. The simpler the installer package, the more likely it is enterprise-friendly. But here are some common pitfalls that can make third-party software unnecessarily difficult to deploy in an enterprise environment.

Pitfall #1: using alternate packaging formats

To make your installer enterprise-friendly, use the Apple Installer package format. This does not mean you have to use Apple's PackageMaker tool to create your packages, but whatever tool you use must have as its final output an installation package compatible with Apple's Installer. If you use any other format for your installer, enterprise administrators

will almost certainly have to repackage your software to deliver it to their users.

This pitfall extends to the so-called "drag-and-drop" disk images: with this type of install, the user is given a disk or disk image, that when opened, presents the software and typically, instructions to drag it into the Applications folder. While this installation method is acceptable for simple interactive installs, it does not work with mass-deployment tools, once again forcing the administrator to repackage your software.

If you (or your customers) prefer drag-and-drop installs and don't want to give them up, at least consider making an Installer package available as an option for system administrator use. Such an optional package could even be included on the disk image that contains the drag-and-drop application, perhaps in a subfolder and mentioned in a READ ME file.

Pitfall #2: asking for administrator credentials on first launch

Some software asks for administrator credentials on first launch so that it can install additional items. Some examples are TextWrangler, and TextMate, each of which ask the user if they can install command-line tools, which requires administrator rights. If the user declines, the software won't ask again. Worse examples include some of the Adobe Acrobat family of products, which ask every time the product is launched until its demands are met.

This is problematic in an enterprise environment because best practices dictate that most users do not have administrative rights, and so cannot provide the requested credentials. The assumption that a user of your software has or can obtain administrator rights after your software has been installed is simply a bad assumption in an enterprise environment, or any other large-scale environment (like education).

To avoid the issue, administrators have to repackage the software; either including the optional installations or packaging a modified version of the software that doesn't ask the user to install the additional items.

From an administrator's viewpoint, it would be better if the installer simply installed everything it wanted installed up front. In the case of TextWrangler and TextMate, there are no installers – these are distributed as drag-and-drop disk images – so again, the option of an Installer package would make administrators lives easier. In the case of software like Acrobat or Reader, vendors should provide a way for administrators to turn this behavior off without having to modify or repackage the software.

Pitfall #3: pre- and post-install scripts

If your installer makes use of pre- and/or post-install scripts, test them during your remote install tests to make sure they don't do anything visible. The Microsoft Office 2008 installer has a post-install script that attempts to add things to the dock. When run remotely when no one was logged in on a Tiger machine, this script caused the Finder and Dock to open behind the login window. Not only was that annoying, it was a security risk since the Finder was running as root! The

Office 2008 installer is not the first installer to do something like this, and unfortunately, probably won't be the last.

Even if a user is logged in, if the administrator was installing Office 2008 remotely, is it really good form to kill and relaunch a user's Dock out from under them?

Other things to watch out for are scripts that attempt to quit open applications, and anything that makes use of AppleScript, which may not behave as expected when no-one is logged in.

Fortunately, a pre- or post- script can easily tell if it's being run in the context of a non-GUI install: the installer command sets the COMMAND_LINE_INSTALL environment variable. Just test for it and exit or skip over a task if it's set – here's a Perl example from a postflight script in the iTunes install package that updates the Dock:

```
# exit if command-line install
exit(0) if ($ENV{'COMMAND_LINE_INSTALL'});
```

Pitfall #4: Licensing and registration

It's a fact of life that many commercial applications need some sort of license code or registration code to run as an effort to combat piracy. If your installer requires entering a license code as part of the install process, it will not be able to be installed via mass-deployment tools. Site licensing can help the problem somewhat, but still doesn't solve it completely if the installer asks for the site license code. Consider these alternatives:

If the software is being installed via command line (i.e. the COMMAND_LINE_INSTALL environment variable is set), allow the installation to occur without entering a registration code. Your app could then ask for the registration code at first launch. (But that can cause its own problems…)

An ever better option, but one that I've seen very few vendors embrace, is to provide administrators with a method to include the registration code or codes with the installer. This could take the form of a separate package install that simply installs the registration code(s), or a properly named and formatted text file that a script included with the installer looks for and uses, or some other method for an admin to provide registration codes without having to visit each machine or do a full repackaging of your software.

There are certainly other failure modes and issues that can make an installer less than enterprise-friendly, but these are among the most common pitfalls.

More information

Apple has some documentation on software delivery, managed installs and remote installs. A PDF is available here:

http://developer.apple.com/documentation/developertools/conceptual/SoftwareDistribution/SoftwareDeliveryGuide.pdf

and an HTML version is here:

http://developer.apple.com/documentation/developertools/conceptual/SoftwareDistribution

Call to action

Software vendors: if you follow the suggestions in this article, your installer will be enterprise-friendly and you may even see increased sales and fewer enterprise support issues.

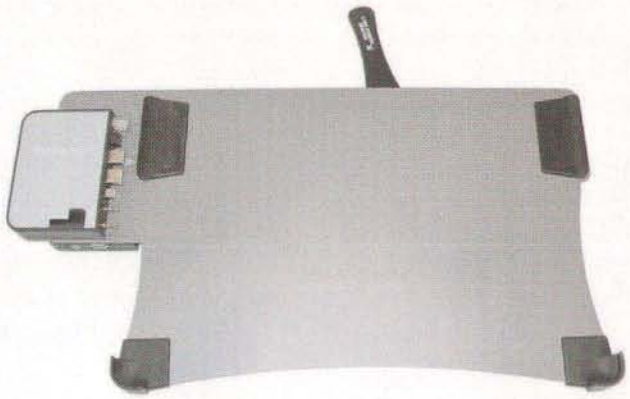
System administrators: if your software vendors aren't providing enterprise-friendly installers, let them know! File bug reports, pester your account representatives, or, worse case, investigate alternatives from vendors who "get it." Good luck!

About The Author

Greg Neagle is a member of the steering committee of the Mac OS X Enterprise Project (macenterprise.org) and is a senior systems engineer at a large animation studio. Greg has been working with the Mac since 1984, and with OS X since its release. He can be reached at gregneagle@mac.com.

BookEndz

by Dennis Sellers



Laptops are great, but those who use them as their primary computer probably connect them to an external monitor, printer and other devices when they're at home or at the office. That can mean lots of peripheral connecting and disconnecting, as well as cable clutter. The BookEndz Docks (http://www.bookendzdocks.com/) are designed to deal with this problem. In fact, BookEndz' product line includes docking stations for Apple laptops dating back to 2000.

As this article was being written, BookEndz was accepting pre-orders for docking stations for Apple's "unibody" MacBooks and MacBook Pros. The station for the 17-inch Unibody MacBook Pro will ship later this year. Pricing is US$159 for 13-inch docking stations, $299.95 for 15-inch docking stations, and $319.95 for 17-inch docking stations.

The docking stations are designed to eliminate cable confusion and damage to connectors. Once the peripheral devices are plugged into the BookEndz Dock, they can remain in and the portable can be connected or disconnected as needed. An "ejection system" ejects the laptop.

For example, the 17-inch docking station replicates all of the connectors and ports from the sides of the 17-inch MacBook Pro and routes the connectors to the back of the dock and manages the new MacSafe Power Connector. The dock allows for all of the peripheral devices normally connected to the sides of the MacBook — Power, Ethernet, Modem, Video, FireWire 400 and 800, USB, Audio In, Audio Out — to be plugged into the back of the dock, and remain plugged in, independent of the computer. Additionally, the dock includes a powered USB hub for a total of five USB ports rather than the standard 2-3 found on Apple's laptops. There are also full-size VGA and DVI ports.

This means that taking the computer from the office or the home, or moving it from one room to another, doesn't require disconnecting and reconnecting several devices. However, you must use your MagSafe power supply to power up your MacBook Pro. The dock doesn't supply power (Apple doesn't license the MagSafe technology to other companies), but it does have a feed-through slot to house the MagSafe Connector.

Using a BookEndz Dock with your laptop involves a few simple steps. First, place the BookEndz Dock on the flat surface on which you'll be using it. Plug all of your external devices into the? station (video monitor, network, printer, etc.).

Shut down your laptop and close the LCD screen.

Move the release handle on the BookEndz Dock to the vertical position. Place the laptop on to the BookEndz Dock. Make sure that the ports of the MacBook Pro align with the connectors of the BookEndz Dock. The connectors mate with those on your laptop and allow you to dock and undock with your external devices. Once those ports are lined up push the left hand side of Docking Station into the ports on the laptop.

Restart your laptop and allow your peripheral devices to boot-up.

When you're ready to travel with your MacBook or MacBook Pro, shut down your laptop and all external devices and close the LCD screen. It's not necessary to unplug them from the dock.

Move the release handle to the vertical position. The BookEndz Dock will undock from your laptop.

The BookEndz's male connectors have a noticeable bit of flexibility, which concerns some folks about their durability. This looseness is designed to allow the connectors to align with their respective female notebook sockets, even sockets that might be slightly off. However, you should certainly dock and undock your laptop with care. Don't go yanking it loose from the BookEndz. Also, the first few times you use the BookEndz, make sure everything is lined up perfectly. After that, it's a little smoother to use.

There have also been comments that the design of the BookEndz is a bit unwieldy. However, considering the very purpose of the docking station, that's probably unavoidable.

After all, the BookEndz' purpose is to "transform" your laptop into a desktop computer. For those who carry a laptop between home and office, two of the docking stations might be a convenience worth paying for. It's a matter of how often you connect and disconnect your Apple laptop from several peripherals and accessories. If you'll be docking and un-docking your laptops several times a day, the price tag will be worth it.

MT

About The Author

Dennis Sellers is a long time journalist. He started in the newspaper business, but has been in the online journalism business for the past 15 years. He's the editor/publisher of Macsimum News (http://www.macsimumnews.com)

Give your iPhone, 3G, 3GS, BlackBerry® or Smartphone more than double the juice.

For iPhone/iPod
Direct plug-in / no cable.
Includes 2G / 3G support brace.
RS001 **$69.95**

For iPhone/iPod with Cable
Cable connection only.
Perfect for iPod touch.
RS008 **$69.95**

For All BlackBerry/Smartphones
Cable or direct plug-in.
Works with all USB-port phones.
RS007 **$69.95**

Built-in super bright LED flashlight.

Built-in laser pointer.

The quick, pocket-sized solution to top up your iPhone, BlackBerry or Palm™ Pre®! And now get a free iPhone case too!

by Richard Thalheimer

Just plug RichardSolo 1800 into your iPhone/BlackBerry /Palm Pre once or twice a day for fifteen minutes, and keep your device charged up. At your desk or at dinner, plug in RichardSolo to instantly transfer charge. No more battery worries! 1800 mAh lithium-ion battery is largest in its class and holds its charge for months. RichardSolo 1800 will charge iPhone to full 1.5 times, and lithium-ion is good for 3-5 years of recharges.

Retractable USB cable, 110-240v AC wall charger, and dual-port USB car charger included — like getting two extra chargers for free! Even charge the RichardSolo 1800 and phone together at the same time. Take only one charger when traveling and wake up in the morning with the RichardSolo and your phone charged. Put the RichardSolo in your pocket, plug it into your phone now and then to top it up. Use your phone while charging it.

Enjoy a built-in laser pointer and ultra-bright LED flashlight, too! These two lights use very little power, so you get handy extra features. RS001 is the only snap-on battery available with a latching mechanism for iPhone, to keep battery firmly attached. RS001 fits all iPhone/iPod cases except the very thickest armor. Optional 2G/3G support brace included.

RS001 works with iPhone 3G and all iPod models except shuffle. Direct plug in. No cable.

RS008 works with iPhone and all iPod models except shuffle — perfect for iPod touch. Cable only — included.

RS007 works with all BlackBerry models, Palm Pre and any phone with a mini/micro USB charging port. Plug in directly or use with cable — included.

Your satisfaction is guaranteed, with our 30-day, no-hassle return privilege. If you're not satisfied for any reason, we'll email you a pre-paid return label. You have no risk whatsoever, and the warranty is now one full year.

On-line ordering and blog reviews: www.RichardSolo.com

email: support@RichardSolo.com

Made for iPod | Works with iPhone

We are here to support you! Actual customer comments:

"To have your company exhibit such exceptional service is unbelievably refreshing." — ***P.S.***

"Dear Richard and Team: This is what I call great customer support. I wish more companies would figure this out these days. Thank you so much." — ***D.C.***

Reviewers give it top marks.

"Now RichardSolo has released a much improved version of the Back-up Battery that not only provides more power for your iPhone or iPod, but some excellent new features that trash the competition."

— ***TUAW.com***

"The quintessential accessory for all iPhone/ iPod owners. If you don't already own one, you need to order yours today."

— ***BuyMeAniPhone.com***

RS001 is also available at BED BATH & BEYOND

RS001 price may vary at BBB. Case not included.

Included Free:

- *AC wall charger*
- *Dual-port USB car charger*
- *Retractable USB cable*
- *Support brace for iPhone*
- *Cable shown with model*

Plus

When you order online from RichardSolo, you get our **Custom Case for iPhone 3G/3GS included** with your RS001 or RS008!

Good protection, textured finish, works perfectly with our support brace.

Order now online: www.RichardSolo.com

Richard|Solo®

Catching up with Screen Capturing

Four products to help you seize the moment, and more.

by Dennis Sellers

Camtasia for Mac

There are lots of screen capture apps available for the Mac, but TechSmith Corp. describes its new Camtasia for Mac as screen-recording, or screencasting, software. It's designed to let you create presentations, demonstrations, screencasts, training and marketing videos in HD-quality video.

Camtasia for Mac records what you see on your Mac's screen, as well as what you say, and how you interact with any web site or Mac-based application such as Keynote and PowerPoint. You can then edit the content and share it online, on http://Screencast.com, on CDs or DVDs, or via iTunes for playback on Apple devices such as the iPhone, iPod touch and iPod.

You can combine real-world footage with screen-recorded content, including webcam and DV video camera footage to create picture-in-picture, picture-then-picture and even side-by-side presentations. Final Cut users can also combine HDV camera footage.

Camtasia for the Mac includes TechSmith's patented SmartFocus technology, so you'll be able to deliver your video at any size while still maintaining the highest viewing quality whether it's for your blog, website, or an Apple mobile device. You can add multiple audio tracks, and virtually unlimited video tracks to layer on images and additional video.

The multiple audio and video tracks can be moved and edited independently in Camtasia for the Mac. If you want to create dynamic screencasts with lots of screen action, or want to highlight a particular section, or create a fly-over effect, Camtasia for the Mac allows you to add and edit SmartFocus key frames for greater precision over zoom and pans, as well as edit key frames of any element on the timeline and change its location and properties over time.

You can add text annotations, shapes and arrows to your screencasts to draw viewers' attention to a particular area while providing more instruction, and add title and credit clips throughout to brand videos before sharing. Camtasia for the Mac produces files with all the options that QuickTime currently supports, including the playback of H.264 MOV videos in the Flash player.

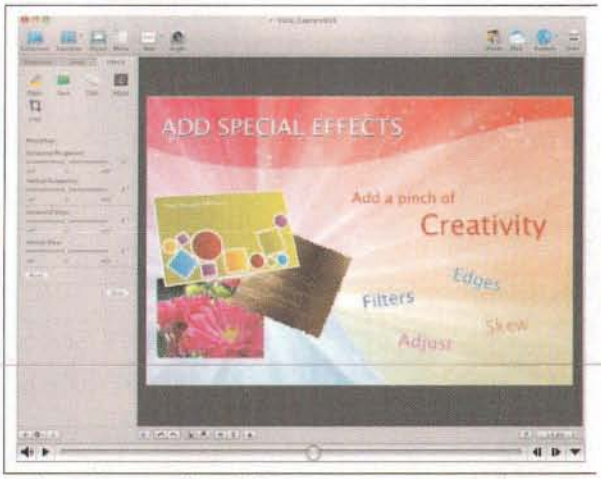
http://Screencast.com

Screencast.com is TechSmith's hosting service for video sharing. Camtasia for Mac trial users and customers receive free 2GB of storage and 2GB of monthly bandwidth. Similarly, Camtasia for Mac includes the ability to share files to YouTube with YouTube account integration.

Camtasia for Mac supports Mac OS X 10.5.6 or later, including support for Mac OS X 10.6 ("Snow Leopard"). Web cameras and microphones for capturing audio and video are available for purchase separately. After the special introductory price of $99 (being offered for a limited time), Camtasia for Mac will have a suggested retail price of $149 with a 30-day free trial, including free training videos and technical support. The software and recommended hardware can be purchased at <http://www.techsmith.com>.

Voila

In May Global Delight <http://www.globaldelight.com/> released Voila, a new image capturing, annotating and sharing tool that boasts the ability to capture a region from the screen in rectangular, circular, polygonal or free hand form.



The software offers full screen capture, timed capture or menu capture. In addition, Voila is integrated with a blend of

annotating tools like texts, arrows, callouts, sprays, stamps and more. You can share your work via email or iPhoto. The software supports web page and webcam captures, an organizer for image management and additional annotations like blur and flip.

Voila can also capture menus, objects, fullscreen or timed captures; you can capture these shapes simultaneously. What's more, you can rotate, resize, switch between captures or undo a selection before capturing.

Let's say you want to capture a web page. Type in the ULR of the page into Voila's built-in browser, and you can snap the current Safari page with a click. Or you can import the web address to snap without leaving your browser. You can browse any web page using the built-in browser and capture the DOM elements. Voila allows you to select more than one DOM element at a time within a web page and snap all of them simultaneously. (DOM, or Document Object Model, is a cross-platform, language-independent convention for representing and interacting with objects in XHML, XHTML and XML documents.)

Global Delight built Voila to work with all the Mac's features. For example, you can capture from your Mac's built-in iSight camera or any third party web cams attached to your computer. You can take a single snap or multiple snaps and add a variety of elements to it using annotation tools like Text, Arrow, Line, Shape, Callouts, Blur, Brush and Stamp. All the annotations are selectable and can be scaled to just about any size. If you later revisit the annotations, you can select, rotate, resize, reorganize or copy-paste across other images.

Voila lets you add metadata to your snaps. You can add titles, relevant tags and describe the information for each of the snaps.

One the most appealing things about Voila is that you can capture multiple screenshots at once. You select different portions of the screen, then adjust, rotate or undo a selection. And — voila! (pun intended) — you can snap simultaneously.

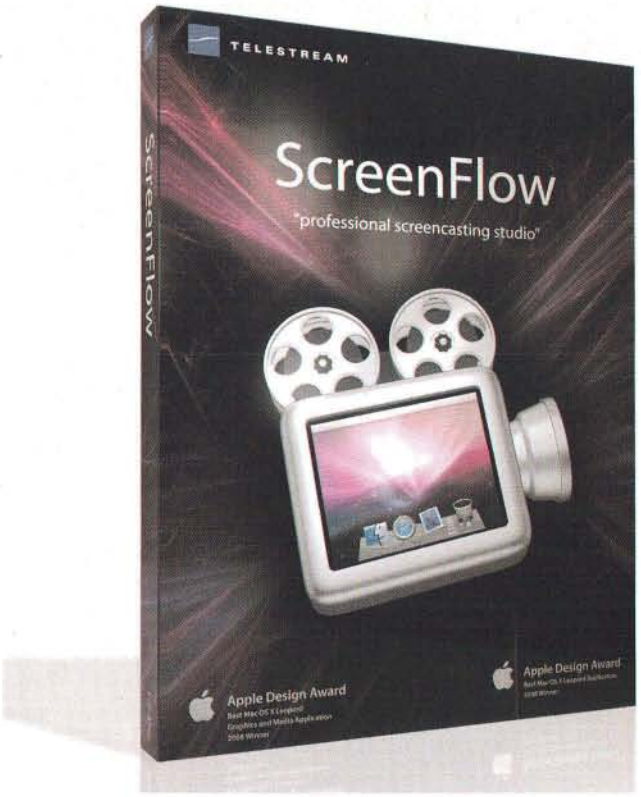
The software sports global keyboard shortcuts to capture full screens, area, windows, web pages, menus and selection captures. You change the keyboard shortcuts in the Voila preferences to meet you needs.

Voila saves all your snaps76 in a central image library and helps you retrieve your creative works at a later date. If you want to group similar snaps, you can create custom collections and drag images into them.

Voila runs on Mac OS X 10.5.2 or later, and is designed to run on any Intel-based Mac running at 1.25GHz and above. iPhoto 6.0.1 or later is required for exporting captured or annotated images.

ScreenFlow

Sometimes it comes in handy to make a screen recording of something you're doing on your Mac so you can show it to someone else. Enter ScreenFlow from Telestream (< www.telestream.net/screenflow>, a US $99.99 "professional screencasting studio" for Mac OS X 10.5 . Apple gave it two Apple Design Awards in 2008: Best Mac OS X Leopard Application and Best Mac OS X Leopard Graphics & Media Application.



Who needs the app? Let's count the ways — or rather the people.

One: those who need to produce demonstrations of software applications. Developers can record a demonstration of their software with audio and even separate video. They can highlight areas of interest, add text, then edit the whole thing and save it as a video. iPhone developers can also take advantage of the utility. With ScreenFlow, they have a tool to record their applications (using any iPhone simulator), and can choose a finger as a mouse pointer as they navigate around their iPhone app.

Two: teachers who want to record their lectures, tutorials, or instructions for students, or to create rich-media presentations. Educators can record their message once, and students can access the video at any time.

Three: those who need to record training sessions. For example, corporate trainers can combine a screen recorded video with Keynote or PowerPoint presentations and video conferencing.

With ScreenFlow you can record from your computer desktop, a video camera, microphone and computer audio simultaneously. Toss in the editing functions, and the finished result is a QuickTime or Windows Media movie, ready for publishing to your web site or blog.

With ScreenFlow you don't have to select an area of the screen for capture, as it sports advanced algorithms that only encode areas of change on your screen. During your screen capture ScreenFlow tracks where your mouse cursor is, when

you click and when you press a key. This allows you to add mouse click effects (both visual and audible) and an overlay showing your key strokes. You can even you zoom the mouse pointer up and down.

"Callouts" let you highlight and focus in on the mouse or frontmost window. For example, want to circle the area around the mouse? You can do it.

ScreenFlow captures are instant. When you stop capturing, you can start editing. You only have to wait when you're exporting the final movie, as this requires processing.

Once your screen capture is complete, you can edit it using a timeline interface that lets you add zoom and pan effects, trim clips, add drop shadow and reflection, adjust audio levels and more. ScreenFlow introduces actions to the editing interface for modifying the parameters of your screencast over time. For example, adding a video action lets you put zoom and pan effects on your clips, while the audio action lets you adjust volume at different points in your screencast. You can also combine existing media into your screencast.

The latest version (1.5) of ScreenFlow adds the ability to add text objects to the timeline, allowing users to add titles to videos. It also adds 10 new audio effects that can be applied to the narration. This is the version that adds the ability to product screen recordings in a Windows Media video format. Direct Media Export requires a Flip4Mac WMV Studio license for $49. Telestream has announced that version 2.0 will be arriving right around the time this issue appears.

When resizing high resolution screen content into a QuickTime movie, ScreenFlow uses custom GPU algorithms to give your finished movie the best possible quality. The software takes advantage of such Mac OS X technologies as Core Animation, QuickLook, Spotlight, QTKit, Quartz Composer, OpenGL, Core Image, Automator, Core Data and many others.

ScreenFlow is Universal Binary so runs natively on both PowerPC and Intel Macs. It's a Leopard only product. A demo is available for download.

Snapz Pro X

Snapz Pro X <http://www.ambrosiasw.com/utilities/snapzprox/> from Ambrosia Software allows you to record anything on your screen, saving it as a QuickTime movie or screenshot that can be emailed, put up on the web or distributed as you wish. Think of it as Grab (the screen capture software that comes with Mac OS X) on steroids.

The software is designed for those who need to make training videos, produce product demos, create tutorials, write documentation, archive streaming video and more. In other words, it captures full motion video screens, not just static screenshots (as Grab does).

Desktop video capture utilities generally come in two varieties. Some can save video recordings of your desktop operations as you're doing them. Others save everything into memory, then postprocess the creation of the final output video files once your session is complete.

Snapz Pro X uses the second method of capture, which has the advantage of postprocessing the file, eliminating the possibility of not recording the smooth playing of video clips in your capture session. Of course you do have to wait for the video to render after you're done recording.

Once your recording is complete, you choose the video-compression format from your QuickTime-compatible codecs, along with frame rate, color depth, data rate and audio-compression settings. You even see a real-time preview of the movie as it's being rendered.

By default, Snapz's hotkey is Command-Shift-3. When invoked, Snapz appears and presents four ways to capture: entire screen, objects (such as menus and dialog boxes), selection, and movie.

The Ambrosia app supports digital audio and supports optional microphone voiceover. Snapz Pro X can save screenshots as .bmp, .pict, .gif, .jpg, .png, .tiff, .pdf, or Photoshop files with control over image compression. Screenshots can be scaled, cropped, color depth-changed, and dithered. Snapz Pro X can also add borders, generate automatic thumbnails, overlay watermarks/copyright notices and more.

Snapz Pro X costs US$69; a demo version is available for download. The software is Universal Binary so runs natively on both PowerPC and Intel Macs. Snapz Pro X requires Mac OS X 10.5 or later. A Mac running with at least an 1.25GHz G4 processor and sporting 1GB of RAM is recommended.

There's also a $29.95 version which lacks the movie capture features. It offers still image capturing only.

MT

About The Author

Dennis Sellers is a long time journalist. He started in the newspaper business, but has been in the online journalism business for the past 15 years. He's the editor/publisher of Macsimum News (http://www.macsimumnews.com)

Advertiser/Product Index

AcctVantage....32
AMS Rabbit....48
Appriver....9
Axiotron, Inc....31
Benchmark Email....43
Citrix Online Division....1
CodeWeavers, Inc....42
Cognito....35
CSO Corp....4
Da-Lite Screen Company, Inc....46
Dexim....IFC
EMC Retrospect....BC
eSellerate/MindVision....29
Etymotic Research, Inc....51
Faronics Corporation....18
Fontlab Ltd....59
Future Media Concepts....44
GeoVid....21
Global Delight Technologies Private Ltd....63
Go4Cast, Inc....47
IDG World Expo Corporation....11
IGC, Inc. / MaxEMail.com....52
Intego, Inc....13
JAMF Software LLC....10
Just Mobile Ltd....58
KAE Corporation....73
Kerio Technologies Inc....66
LassoSoft LLC....53
LC Technology International, Inc....24, 71
Lemke Software GmbH....34
Limit Point Software....67
LithiumCorp....69
MacResource Computers & Service....64
MacsDesign Studio LLC....65
MacSpeech, Inc....19
Micromat, Inc....39
Microsoft....57
Monotype Imaging....37
Mosso :: The Hosting Cloud....45
Now Software....27
OlympicControls Corp....23
On-Target Reports, Inc....12
Parallels Inc....2-3
Pixologic....72
Powerbookmedic.com....33
PrintDirectforLess.com, Inc....25
REAL Software, Inc....49
RichardSolo....75
Seapine Software, Inc....55
Small Dog Electronics....IBC
Small Tree Communications....62
SmileOnMyMac, LLC....17
Smith Micro Software, Inc....41
SuperSync....67
TechSmith Corporation....15
Telestream....48
TrueToniqs....24
Universe Software GmbH....34
Worldwide Media Inc. - MostWantedDomains....16
Yazsoft.com....26
ZAGG Inc....47

AcctVantage • AcctVantage....32
Application Lifecycle Management (ALM) • Seapine Software, Inc....55
Benchmark Email • Benchmark Email....43
BookEndz • OlympicControls Corp....23
Brain Toniq • TrueToniqs....24
Camtasia • TechSmith Corporation....15
Casper • JAMF Software LLC....10
ConceptDraw • CSO Corp....4
CD/DVD duplication • AMS Rabbit....48
CrossOver • CodeWeavers, Inc....42
Da-Lite Screen • Da-Lite Screen Company, Inc....46
Deep Freeze • Faronics Corporation....18
Domain Services • Worldwide Media Inc. - MostWantedDomains....16
eSellerate • eSellerate/MindVision....29
Exchange Hosting • Appriver....9
Font Editor • Fontlab Ltd....59
FontExplorerX • Monotype Imaging....37
GoToMeeting • Citrix Online Division....1
GraniteSTOR • Small Tree Communications....62
Graphic Converter • Lemke Software GmbH....34
In-Ear Technology • Etymotic Research, Inc....51
invisibleSHIELD by ZAGG • ZAGG Inc....47
iPhone Batteries • RichardSolo....75
iPhone Battery / Dock • Dexim....IFC
IT Training • Future Media Concepts....44
Kerio Server Software • Kerio Technologies Inc....66
Lasso • LassoSoft LLC....53
Lithium Network Monitoring • LithiumCorp....69
Mac Security Bundle • Intego, Inc....13
MacResource Computers • MacResource Computers & Service....64
MacSpeech Dictate • MacSpeech, Inc....19
Macworld Expo • IDG World Expo Corporation....11
maxemail.com • IGC, Inc. / MaxEMail.com....52
ModBook • Axiotron, Inc....31
MoneyWorks • Cognito....35
Mosso • Mosso :: The Hosting Cloud....45
Now Up-to-Date • Now Software....27
Office 2008 for Mac • Microsoft....57
On-Target Reports • On-Target Reports, Inc....12
Parallels Desktop and Server • Parallels Inc....2-3
PDF Office • Universe Software GmbH....34
PDFpen • SmileOnMyMac, LLC....17
PHOTORECOVERY®/FILERECOVERY® • LC Technology International, Inc. .24, 71
Powerbookmedic.com • Powerbookmedic.com....33
Printing Services • PrintDirectforLess.com, Inc....25
Protemac • GeoVid....21
Rack Mounts • KAE Corporation....73
REALbasic • REAL Software, Inc....49
Retrospect • EMC Retrospect....BC
ScreenFlow • Telestream....48
SmallDog.com • Small Dog Electronics....IBC
Speed Download • Yazsoft.com....26
StuffIt • Smith Micro Software, Inc....41
SuperSync • SuperSync....67
TechTool Pro 5 • Micromat, Inc....39
Training / Consulting • Go4Cast, Inc....47
Utilities • Limit Point Software....67
Voila Screen Capture • Global Delight Technologies Private Ltd....63
WebHelpDesk • MacsDesign Studio LLC....65
Xtand • Just Mobile Ltd....58
ZBrush • Pixologic....72

THE MACTECH SPOTLIGHT

Jesse Grosjean

Hog Bay Software

http://www.hogbaysoftware.com

What do you do?

Everything at Hog Bay Software.

How long have you been doing what you do?

I've been working full time on Hog Bay Software since 2003. Before that I worked on it in the evenings and weekends, since around 2001.

What was your first computer?

The first real one was a 286 that we got when I was in 8th grade. In high school I helped write a grant (well I don't think I wrote anything, but I acted as an excited student) to get the schools first Mac. I was mostly interested in graphics design then... did a little programming in Macromedia director, but didn't really learn programming until college.

Are you Mac-only, or a multi-platform person?

Mac only.

What is a utility that you use daily for your work?

I spend 90% of my time in Xcode and Safari. My favorite third party app is DeskShade... my desk is a mess, but I like to pretend that it's not.

What is the advice you'd give to someone trying to get into this line of work today?

I've also found it very useful to try to write a tagline and an elevator pitch for any project. That's before I start coding. For example with TaskPaper my pitch is:

"For Mac users who want an easier way to make lists and stay organized. TaskPaper is a simple to-do list that's surprisingly adept. Unlike today's complex organizers, TaskPaper lets you focus on getting things done."

It might sounds a little cheesy, but the goal isn't to turn yourself into a salesman, instead the goal is to get your project focused. If you can't describe your project in terms of an elevator pitch then it's probably not focused enough.

What's the coolest tech thing you've done using OS X?

I wrote Piccolo, a Java framework for zooming user interfaces (ZUI's) http://www.cs.umd.edu/hcil/piccolo/. It was pretty cool, though it's showing its age a bit now. Right now my coolest app is TaskPaper. It looks really simple, but I think its basic metaphor is quite powerful and will serve as the base for my future projects. Plain text data, with parsed out structure at runtime.

Where can we see a sample of your work?

I do everyting at http://www.hogbaysoftware.com. I also just launched http://www.writeroom.ws.

The next way I'm going to impact IT/OS X/the Mac universe is:

In the near term I don't see any new products, just new versions of WriteRoom and TaskPaper. Long term I have big ideas, but no elevator pitch yet, so I'm not focused enough and shouldn't try describing them. You'd just see lots of excited hand-waving from a techie.

MT

If you or someone you know belongs in the MacTech Spotlight, let us know! Send details to editorial@mactech.com

Mac shopping made easy.

Grab that to-do list, and prepare for some one-stop shopping at Smalldog.com!

Bundles simplify the buying process

Mac bundles (think Mac + RAM + AppleCare + external hard drive, etc.) not only include **everything you need,** but also **save you money.**

Visit » **Smalldog.com/specials**

Macs from under $500

We carry all **current Macs** as well as **used, refurbished and closeout models,** so there is a Mac for any budget.

Visit » **Smalldog.com/macs**

Free shipping over $200

It's true–we provide **free, same-day ground shipping** on every item over $200 every day.

Tax-free shopping

Purchases outside of Vermont are always shipped **tax-free.**

✓ 13" MacBook Pro + Chill Pill® mobile speakers

Small Dog Electronics

Always By Your Side

www.smalldog.com

800-511-MACS

Apple Specialist

Celebrating 15 Years • 3rd Largest Apple Specialist in New England • 5-Star Merchant Rating • Same-day shipping

Bundles

Macs

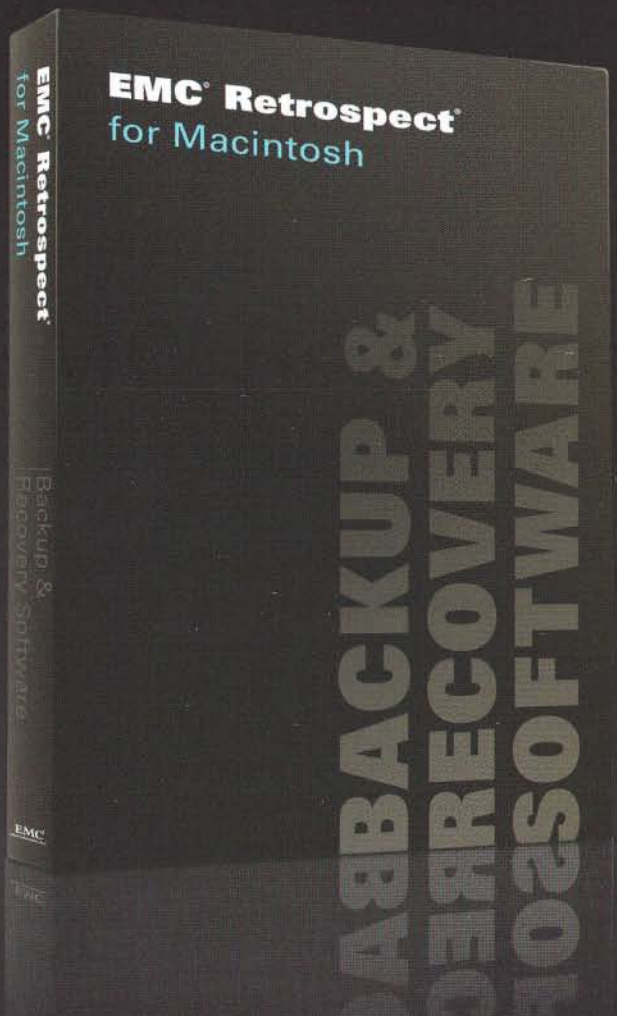
Free Shipping

Tax-Free

NEW!

EMC® Retrospect® 8

backup and recovery software for small and medium businesses



The most trusted name in Mac backup

Now available for PowerPC & Intel Macs!

All-new EMC Retrospect 8 for Macintosh provides the reliability, ease of use, power, and flexibility you need to protect critical data on Mac and Windows PCs and servers. EMC Retrospect includes a state-of-the-art Mac user interface and enterprise-level features — including remote management of one or more backup servers, disk-to-disk-to-*anything* backups, Xsan support and custom reporting — at a fraction of the cost of other products.

Download a free 45-day trial at **www.retrospect.com/wwdc**

EMC²
where information lives